

Evaluierung bedeutender Webapplikations- Entwicklungsframeworks im Hinblick auf die Aspekte Rich-Client-Funktionalität und Barrierefreiheit im Kontext der Anforde- rungen im Bereich eGovernment

**Walter Kern
Universität Regensburg
Philosophische Fakultät IV:
Sprach- und Literaturwissenschaften**

11. August 2008

Alle genannten und gegebenenfalls durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichnungsrechts und den Besitzrechten der jeweiligen Eigentümer. Allein aufgrund der bloßen Nennung kann nicht geschlossen werden, dass Markenzeichen nicht durch Rechte Dritter geschützt sind. Aus dem Fehlen des Zeichens (R) darf nicht geschlossen werden, dass ein Name oder Zeichen frei ist. Eine Haftung für ein etwaiges Fehlen des Zeichens (R) wird ausgeschlossen.

Evaluierung bedeutender Webapplikations-Entwicklungsframeworks im Hinblick auf die Aspekte Rich-Client-Funktionalität und Barrierefreiheit im Kontext der Anforderungen im Bereich eGovernment

Vorwort

Im Rahmen dieser Evaluierung sollen die vielversprechendsten¹ und bedeutendsten² Frameworks im Web-Umfeld auf die Erfüllung zentraler Anforderungen an ein modernes Webapplikations-Entwicklungsframework mit Anwendungsbereich eGovernment untersucht werden, wobei der Fokus dieser Evaluierung auf den Aspekten Barrierefreiheit und Rich Client-Funktionalität der Komponenten der untersuchten Frameworks liegt.

Hierbei ist zu berücksichtigen, dass Webentwicklung im eGovernment-Bereich in Bayern primär basierend auf .NET- und Java-Technologie stattfindet, weshalb der Fokus der Untersuchung auf Frameworks für diese Plattformen liegt. Zu beachten ist, dass die Betrachtung vorliegender Funktionalität des jeweiligen Frameworks stets auf die reguläre Unterstützung der entsprechenden Funktionalität erfolgt, da Workaround-Lösungen nicht der Maßstab für eine tragfähige Lösung bilden sollen.

Zudem soll Beachtung finden, dass die Möglichkeit besteht, einige der untersuchten Frameworks miteinander zu kombinieren, um einen höheren Grad an Anforderungserfüllung zu erreichen. Aufgrund der übermäßig großen Anzahl an Permutationen soll in dieser Untersuchung eine Beschränkung auf einzelne Technologien oder alternativ bei sehr starker Integration mehrerer Frameworks die am besten aufeinander abgestimmte Lösung untersucht werden.

Schlagwörter: World Wide Web 2.0, Electronic Government, Rich client, Barrierefreiheit, Evaluation, BITV

¹ Primärkriterium ist die Innovativität bzw. Grundsätzlichkeit eines Ansatzes, aber auch Wachstumspotentiale.

² Primärkriterien sind Marktdurchdringung des Produkts, Wachstumschancen und die Marktmacht des Herstellers.

Inhalt

1	PRÄAMBEL.....	1
2	BESTIMMUNG ZU EVALUIERENDER KOMPONENTEN.....	2
2.1	INTRODUKTION	2
2.2	WINDOWS VISTA USER EXPERIENCE GUIDELINES	2
2.3	WINDOWS USER EXPERIENCE GUIDELINES (FÜR WIN. 2000/XP).....	3
2.4	KDE 3 STYLEGUIDE.....	3
2.5	GNOME HUMAN INTERFACE GUIDELINES 2.0	4
2.6	APPLE HUMAN INTERFACE GUIDELINES	4
2.7	SCHNITTFELDBETRACHTUNG	5
3	ANFORDERUNGEN.....	7
3.1	PRÄAMBEL.....	7
3.2	ANFORDERUNGSSPEZIFIZIERUNG.....	7
3.3	ABBILDUNG DER DETAILANFORDERUNGEN AUF DIE UNTERSUCHUNGSGEGENSTÄNDE	8
3.3.1	BESCHREIBUNG	8
3.3.2	RICH-CLIENT-FUNKTIONALITÄT	8
3.3.2.1	Dialoge	8
3.3.2.2	Registerkarten	9
3.3.2.3	Menüs.....	10
3.3.2.4	Symbolleisten.....	10
3.3.2.5	Hierarchische Steuerelemente (Tree Views).....	11
3.3.2.6	Fortschrittsanzeige	11
3.3.2.7	Infotips	12
3.3.3	FUNKTIONSWEISE BEI NICHT-VERFÜGBAREM JAVASCRIPT	12
3.3.4	BARRIEREFREIHEIT NACH BITV	12
3.3.5	BROWSERINTEROPERABILITÄT	12
3.4	RAHMENPARAMETER	13
4	PRODUKTAUSWAHL UND BEWERTUNG	14
4.1	ÜBERBLICK.....	14
4.2	BEWERTUNG.....	15
4.3	GEWICHTUNGSFAKTOREN	15
5	EVALUIERUNG.....	17
5.1	PRÄAMBEL.....	17
5.1.1	BASISPLATTFORMEN- UND FRAMEWORKS	17
5.1.2	EVALUIERTE PRODUKTVERSIONEN	17
5.2	ASP.NET 3.5 (STANDARD)	18
5.2.1	BESCHREIBUNG	18
5.2.2	EVALUIERTE PRODUKTVERSION	18
5.2.3	DETAILUNTERSUCHUNG.....	18
5.2.3.1	Dialoge	18
5.2.3.2	Registerkarten	18
5.2.3.3	Menüs.....	18
5.2.3.4	Symbolleisten.....	21

5.2.3.5	Hierarchische Steuerelemente (Tree Views)	22
5.2.3.6	Fortschrittsanzeige	25
5.2.3.7	Infotips	29
5.2.4	FAZIT	29
5.3	AJAX CONTROL TOOLKIT	29
5.3.1	BESCHREIBUNG	29
5.3.2	EVALUIERTE PRODUKTVERSION	29
5.3.3	DETAILUNTERSUCHUNG	29
5.3.3.1	Dialoge	29
5.3.3.2	Registerkarten	34
5.3.3.3	Menüs	38
5.3.3.4	Symbolleisten	42
5.3.3.5	Hierarchische Steuerelemente (Tree Views)	42
5.3.3.6	Fortschrittsanzeige	42
5.3.3.7	Infotips	42
5.3.4	FAZIT	45
5.4	COMPONENTART WEB.UI	46
5.4.1	BESCHREIBUNG	46
5.4.2	EVALUIERTE PRODUKTVERSION	46
5.4.3	DETAILUNTERSUCHUNG	46
5.4.3.1	Dialoge	46
5.4.3.2	Registerkarten	50
5.4.3.3	Menüs	54
5.4.3.4	Symbolleisten	57
5.4.3.5	Hierarchische Steuerelemente (Tree Views)	60
5.4.3.6	Fortschrittsanzeige	64
5.4.3.7	Infotips	64
5.4.4	FAZIT	67
5.5	NETADVANTAGE FOR ASP. NET	67
5.5.1	BESCHREIBUNG	67
5.5.2	EVALUIERTE PRODUKTVERSION	68
5.5.3	DETAILUNTERSUCHUNG	68
5.5.3.1	Dialoge	68
5.5.3.2	Registerkarten	72
5.5.3.3	Menüs	76
5.5.3.4	Symbolleisten	80
5.5.3.5	Hierarchische Steuerelemente (Tree Views)	83
5.5.3.6	Fortschrittsanzeige	86
5.5.3.7	Infotips	87
5.5.4	FAZIT	87
5.6	Telerik RADCONTROLS FOR ASP.NET AJAX	87
5.6.1	BESCHREIBUNG	87
5.6.2	EVALUIERTE PRODUKTVERSION	87
5.6.3	DETAILUNTERSUCHUNG	87
5.6.3.1	Dialoge	87
5.6.3.2	Registerkarten	92
5.6.3.3	Menüs	95
5.6.3.4	Symbolleisten	99
5.6.3.5	Hierarchische Steuerelemente (Tree Views)	102
5.6.3.6	Fortschrittsanzeige	105
5.6.3.7	Infotips	106
5.6.4	FAZIT	109
5.7	DEVELOPER EXPRESS ASPXPERIENCE SUITE	109
5.7.1	BESCHREIBUNG	109
5.7.2	EVALUIERTE PRODUKTVERSION	110
5.7.3	DETAILUNTERSUCHUNG	110

5.7.3.1	Dialoge	110
5.7.3.2	Registerkarten	114
5.7.3.3	Menüs.....	118
5.7.3.4	Symbolleisten.....	121
5.7.3.5	Hierarchische Steuerelemente (Tree Views).....	125
5.7.3.6	Fortschrittsanzeige	125
5.7.3.7	Infotips	125
5.7.4	FAZIT	128
5.8	JSF (STANDARD)	129
5.8.1	BESCHREIBUNG	129
5.8.2	EVALUIERTE PRODUKTVERSION	129
5.8.3	DETAILUNTERSUCHUNG.....	129
5.8.3.1	Dialoge	129
5.8.3.2	Registerkarten	129
5.8.3.3	Menüs.....	129
5.8.3.4	Symbolleisten.....	130
5.8.3.5	Hierarchische Steuerelemente (Tree Views).....	130
5.8.3.6	Fortschrittsanzeige	130
5.8.3.7	Infotips	130
5.8.4	FAZIT	130
5.9	MYFACES TOMAHAWK	130
5.9.1	BESCHREIBUNG	130
5.9.2	EVALUIERTE PRODUKTVERSION	131
5.9.3	DETAILUNTERSUCHUNG.....	131
5.9.3.1	Dialoge	131
5.9.3.2	Registerkarten	135
5.9.3.3	Menüs.....	139
5.9.3.4	Symbolleisten.....	143
5.9.3.5	Hierarchische Steuerelemente (Tree Views).....	147
5.9.3.6	Fortschrittsanzeige	150
5.9.3.7	Infotips	151
5.9.4	FAZIT	154
5.10	JBOSS RICHFACES.....	154
5.10.1	BESCHREIBUNG	154
5.10.2	EVALUIERTE PRODUKTVERSION	155
5.10.3	DETAILUNTERSUCHUNG.....	155
5.10.3.1	Dialoge	155
5.10.3.2	Registerkarten	160
5.10.3.3	Menüs.....	163
5.10.3.4	Symbolleisten.....	167
5.10.3.5	Hierarchische Steuerelemente (Tree Views).....	171
5.10.3.6	Fortschrittsanzeige	175
5.10.3.7	Infotips	178
5.10.4	FAZIT	182
5.11	ORACLE ADF FACES RICH CLIENT	182
5.11.1	BESCHREIBUNG	182
5.11.2	EVALUIERTE PRODUKTVERSION	183
5.11.3	DETAILUNTERSUCHUNG.....	183
5.11.3.1	Dialoge	183
5.11.3.2	Registerkarten	188
5.11.3.3	Menüs.....	192
5.11.3.4	Symbolleisten.....	196
5.11.3.5	Hierarchische Steuerelemente (Tree Views).....	199
5.11.3.6	Fortschrittsanzeige	203
5.11.3.7	Infotips	207
5.11.4	FAZIT	210

5.12	NETADVANTAGE FOR JSF	211
5.12.1	BESCHREIBUNG	211
5.12.2	EVALUIERTE PRODUKTVERSION	211
5.12.3	DETAILUNTERSUCHUNG	211
5.12.3.1	Dialoge	211
5.12.3.2	Registerkarten	216
5.12.3.3	Menüs	219
5.12.3.4	Symbolleisten	223
5.12.3.5	Hierarchische Steuerelemente (Tree Views)	226
5.12.3.6	Fortschrittsanzeige	230
5.12.3.7	Infotips	230
5.12.4	FAZIT	231
6	FAZIT	232
	LITERATURVERZEICHNIS	233
	ABBILDUNGSVERZEICHNIS	237
	ANLAGEN	239

1 Präambel

Zielsetzung dieser Arbeit ist der Vergleich der bedeutendsten³ und vielversprechendsten⁴ Kandidaten im Bereich Rich Client-Webapplikationsentwicklungsframeworks bzw. Web 2.0-Entwicklungsframeworks hinsichtlich der Erfüllung von Anforderungen aus der Domäne eGovernment (vgl. Gisler et al., 2001), weshalb die Anforderungen sich neben der Forderung moderner flexibler Software-Architektur (vgl. Vogel et al., 2005) primär an staatlichen Vorgaben, formuliert in Dokumenten wie KBSt (2006a) oder KBSt (2006b), orientieren.

Primär sollen dabei unter Anderem Anforderungen aus dem Bereich Barrierefreiheit eine zentrale Rolle spielen, da Barrierefreiheit entsprechend der Barrierefreien Informationstechnik-Verordnung, kurz BITV (vgl. Bundesministerium des Innern, 2002), oder auch der bayerischen Umsetzung BayBITV (vgl. Bayerisches Staatsministerium des Innern, 2003) als Grundlage jeder Neuentwicklung im Bereich Webapplikationen vorzusehen ist, aber abgeschwächt auch für Altverfahren langfristig Berücksichtigung finden muss.

Zusammengefasst geht es darum, RIA-Frameworks (vgl. Moritz, 2008; Linaje et al., 2007; W3C, 2008) im ASP.NET- und Java EE-Umfeld im Hinblick auf die Erfüllung von Barrierefreiheitsanforderungen, formuliert in der (Bay)BITV, und Anforderungen an moderne Softwareentwicklung, formuliert in Dokumenten wie Vogel et al. (2005), zu untersuchen.

³ Kriterien: Marktmacht des Herstellers, Marktdurchdringung.

⁴ Kriterien: Innovativität, Generizität, Standard-Konformität.

2 Bestimmung zu evaluierender Komponenten

2.1 Introduction

In diesem Kapitel erfolgt eine Betrachtung der zentralen Standard-Steuerelemente⁵ der bekanntesten und verbreitesten Betriebssysteme bzw. von deren Window-Manager-Systeme. Im Anschluss daran soll eine gemeinsame Teilmenge der essentiellen Komponenten aller beleuchteten Betriebssysteme bzw. Window-Manager-Systeme bestimmt werden.

Es wird darauf hingewiesen, dass die jeweils neben einer Komponente vorzufindende Beschreibung stets nur einen grobe Vorstellung der Komponente unterstützen soll. Für eine ausführlichere Darstellung der genannten Komponenten soll auf die angeführten Guidelines verwiesen werden.

2.2 Windows Vista User Experience Guidelines

Nach Microsoft Corporation (2007a) zählen laut den Windows Vista User Experience Guidelines⁶ folgende Komponenten zu den regulären Steuerelementen (sogenannte Common Controls) bei der Desktopentwicklung:

- Balloons (Sprechblasen für Informationsmeldungen)
- Check Boxes (Kontrollkästchen, Multi-Select)
- Command Buttons (Befehlsschaltflächen)
- Command Links (Befehlslinks)
- Drop Down Listen (Dropdownlisten)
- Combo Boxen (Kombinationsfelder)
- Groupboxen (Rahmen-Steuerelemente)
- Links (Reguläre Links)
- Listboxen (Listboxen)
- ListView (ListView-Komponenten)
- Progress Bars (Fortschrittsbalkensteuerelemente)
- Progressive Disclosure Controls (Expandierungs-/Kollabierungselemente)
- Radio Buttons (Optionsfelder)
- Sliders (Schieberegler)
- Spin Controls (Auf-/Ab-Steuerelemente)
- Tabs (Registerkartenkomponenten)
- Textboxen (Textfeldkomponenten)
- Tooltips (Tooltip-Komponenten mit Funktionsbeschreibungen)
- Infotips (Informationsmeldungs-Komponenten)
- Tree Views (Baum-Komponenten)

Daneben spielen auch Benachrichtungselemente, Statusleisten, Menüs und Symbolleisten eine wichtige Rolle (vgl. Microsoft Corporation, 2007a).

Ferner sind nach Microsoft Corporation (2007a) Dialoge und Assistenten weitere wichtige GUI-Elemente.

⁵ Steuerelement = eine Benutzeroberflächenkomponente, z.B. ein Textfeld.

⁶ Vgl. <http://download.microsoft.com/download/e/1/9/e191fd8c-bce8-4dba-a9d5-2d4e3f3ec1d3/ux%20guide.pdf>.

2.3 Windows User Experience Guidelines (für Win. 2000/XP)

Nach Microsoft Corporation (2007b) zählen laut den „Windows User Experience Guidelines für Windows 2000 und XP⁷ folgende Komponenten zu den regulären Steuerelementen:

- Menues (Menüs)
- Toolbars (Symbolleisten)
- Statusbars (Statuszeilenkomponenten)
- Command Buttons (Befehlsschaltflächen)
- Option Buttons (Optionsfelder, Radio Buttons)
- Check Boxes (Kontrollkästchen)
- List Boxes (List Boxen)
- Single Selection List Boxes (Einfach-Selektierungs-Listboxen)
- Drop-Down List Boxes (Drop-Down-Listboxen)
- Extended- and Multiple-Selection List Boxes (Erweiterte und Mehrfachselektions-Listboxen)
- List View Controls (List View-Steuerelemente)
- Tree View Controls (Baum-Komponenten)
- Text Boxes (Textfelder)
- Rich Text Boxes (Rich Text Box-Komponenten)
- Drop-Down Combo Boxes (Drop Down-Komboboxen)
- Spin Boxes (Spin-Buttons)
- Static Text Fields (Statische Textfelder)
- Shortcut Key Input Controls (Tastenkombinationseingabeelemente)
- Date-Picker (Datumswauswahlkomponente)
- Group Boxes (Rahmen-Steuerelemente)
- Progress Indicators (Fortschrittsbalkensteuerelemente)
- Property Sheet Controls (Property Sheet-Komponenten)
- Scroll Bars (Bildlaufleistenkomponenten)
- Sliders (Schieberegler)
- Tabs (Registerkartenkomponenten)
- Tooltip Controls (Tooltip-Steuerelemente)
- Balloon Tips (Sprechblasentooltips)
- WebBrowser Control (WebBrowser-Steuerelement)

2.4 KDE 3 Styleguide

Im Bereich des K Desktop Environment (KDE) liegen nach KDE-Entwicklerteam (o.J.) als Komponenten unter Anderem folgende Steuerelemente auf Basis des KDE 3-Styleguides⁸ vor:

- dialog (Dialog)
- group box (Rahmen-Steuerelement)
- button (Schaltfläche)
- tab (Tab-Steuerelement)
- listview (ListView-Steuerelement)
- menu (Menü)
- treelist (Baum-Steuerelement)
- edit box (Eingabefeld)

⁷ Vgl. <http://www.ms2.cn/downloads/details.aspx?familyid=B996E1E7-A83A-4CAE-936B-2A9D94B11BC5&displaylang=en>.

⁸ Vgl. <http://developer.kde.org/documentation/standards/kde/style/styleguide.pdf>.

- list box (Listbox)
- combo box (Combobox)
- spin box (Auf-/Ab-Steuerelement)
- check box (Kontrollkästchen)
- option button (Optionsfeld)
- slider (Schieberegler)
- pop up (Popup-Steuerelement)

2.5 GNOME Human Interface Guidelines 2.0

Unter Gnome liegen gemäß der GNOME Human Interface Guidelines in der Version 2.0⁹ nach Benson et al. (2004) unter Anderem folgende Steuerelemente als Komponenten vor:

- Dialogs (Dialoge)
- Menues (Menüs)
- Toolbars (Symbolleisten)
- Text Entry Fields (Ein- und mehrzeilige Textfelder)
- Spin Boxes (Auf-/Ab-Steuerelemente)
- Sliders (Schieberegler)
- Buttons (Schaltflächen)
- Check Boxes (Kontrollkästchen)
- Radio Buttons (Optionsfelder)
- Toogle Buttons (Toogle-Schaltflächen)
- Drop Down Lists (Drop-Down-Listen)
- Drop Down Combination Boxes (Drop-Down-Kombinationsfelder)
- Scrollbars (Bildlaufleisten)
- Lists (Listen)
- Trees (Bäume)
- Tabbed Notebooks (Registerkartensteuerelemente)
- Progress Bars (Fortschrittsbalkensteuerlemenete)
- Status Bars (Statuszeilensteuerelemente)
- Frames (Rahmen-Steuerelemente)
- Separators (Graphische Separatoren, Linien)

2.6 Apple Human Interface Guidelines

Die Apple Human Interface Guidelines für Mac OS X¹⁰ beschreiben neben Empfehlungen für den Design-Prozess und dem zu erzeugenden Benutzererlebnis nach Apple Inc. (2008) auch die Basis-Steuerelemente der bei Mac OS X zum Einsatz kommenden Aqua-Oberfläche. Dabei handelt es sich primär um:

- Menues (Menüs)
- Windows (Fenster)
- Dialogs (Dialoge)
- Panels (Anpassbare Container-Steuerelemente)
- Icons (Symbole)
- Buttons (Schaltflächen)

⁹ Vgl. <http://developer.gnome.org/projects/gup/hig/2.0/hig-2.0.pdf>.

¹⁰ Vgl. <http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/>.

- Radio Buttons (Optionsfelder)
- Check Boxes (Kontrollkästchen)
- Combination Boxes / Popup Menues / Command Pop-Down Menues (Kombinationsfelder)
- Actions Menues (Symbolleistendropdown-Schaltfläche)
- Path Controls (Darstellung hierarchischer Pfade)
- Date Pickers (Komponente zur Datumsauswahl)
- Stepper Controls (Spin-Buttons)
- Textfelder (Eingabe, Suche, ...)
- Tab Views (Registerkartensteuerelemente)
- List Views (List View-Steuerelemente)
- Column Views (mehrspaltige Listenkomponente)
- Group Boxes (Rahmen-Steuerelemente)
- Token Fields (Eingabefeld)
- Split Controls (Splitter)
- Placards (Statusleistenkomponenten)
- Progress Indicators / Level Indicators (Fortschritts- bzw. Statusanzeige)

2.7 Schnittfeldbetrachtung

Bei Durchführung einer Schnittfeldbetrachtung der bisher evaluierten GUI-Oberflächen(elemente) kann festgehalten werden, dass die folgenden, nicht im HTML-Standardumfang enthaltenen, Elemente die Verbreitesten darstellen:

- Fenster
- Dialoge
- Registerkarten
- Menüs
- Symbolleisten
- Statusleisten
- Hierarchische Steuerelemente
- Listen-Steuerelemente (List-Views)
- Spin-Boxen
- Schieberegler
- Fortschrittsanzeige
- Infotips

Eine weitere Einschränkung dieser Liste lässt sich dadurch vornehmen, dass diese Komponenten im Webkontext betrachtet werden und zudem auf ihre HTML-Basiskomponenten zurückgeführt werden. So lassen sich Fenster und Dialoge zusammenfassen, Spin-Boxen auf zwei einfache Buttons reduzieren und Listen-Steuerelemente durch einfache Tabellenstrukturen oder programmatische Schleifen samt Ausgabe von Templates ersetzen. Statusleisten auf der anderen Seite können durch einfache Textausgabe ersetzt werden.

Bei Infotips (vgl. Microsoft Corporation, 2007a) handelt es sich um mehrzeilige, anpassbare Tooltips, was einen erheblichen Unterschied zu herkömmlichen Tooltips, die über das HTML-title-Attribut¹¹ realisiert werden, darstellt, sodass Infotips explizit untersucht werden sollen.

¹¹ Vgl. <http://de.selfhtml.org/html/referenz/attribute.htm#title>.

Schieberegler erfordern die Unterstützung von Drag & Drop, was jedoch JavaScript voraussetzt. Die Verfügbarkeit einer Funktionalität nur im Fall der JavaScript-Unterstützung auf Clientseite wird jedoch, auch mit dem Hintergrund des Aspekts Barrierefreiheit (vgl. Bundesministerium des Innern, 2002), nicht toleriert, weshalb diese Komponente aufgrund fehlender technischer, barrierefreier Machbarkeit und zugleich geringer Relevanz von der Untersuchung ausgeschlossen wird.

Damit verbleibt für die weitere Untersuchung folgende Ergebnismenge:

- Dialoge
- Registerkarten
- Menüs
- Symbolleisten
- Hierarchische Steuerelemente (Tree Views)
- Fortschrittsanzeige
- Infotips

Bei Dialogen soll zusätzlich untersucht werden, ob MessageBox-ähnliche Standarddialoge vom jeweiligen Framework unterstützt werden, was es ermöglichen könnte, die von den Browsern unterstützten, rein clientseitigen alert- und confirm-Abfragen durch eine barrierefreie Lösung zu ersetzen.

3 Anforderungen

3.1 Präambel

Grundlage dieser Evaluierung sind folgende Basisanforderungen und Grundsätze, welche sich primär aus dem Dokument SAGA¹² (vgl. KBSt, 2006a) und der Forderung nach einer modernen flexiblen Software-Architektur (vgl. Vogel et al., 2005; Gamma et al., 1994), die sich auch in den Aspekten Generizität, Flexibilität und Erweiterbarkeit (vgl. KBSt, 2006b) wiederfindet, ableiten:

- Rich-Client-Funktionalität
- Funktionsweise bei nicht-verfügbarem JavaScript
- Barrierefreiheit nach BITV¹³
- Browserinteroperabilität

Diese Kriterien sind dabei im Rahmen dieser Untersuchung gegen jede definierte Basiskomponente jedes Frameworks zu prüfen, wobei im Fall einer Nicht-Unterstützung einer gesamten Komponente jeder einzelne Punkt mit der minimalen Bewertungskennziffer zu versehen ist.

3.2 Anforderungsspezifizierung

In diesem Abschnitt sollen die Prüfkriterien, welche als Basis der Evaluierung aller Frameworks dienen, näher bestimmt werden.

Zunächst soll der Umfang der „Rich-Client-Funktionalität“ geprüft werden. Wenngleich die originäre Funktion der jeweiligen Komponente prinzipiell ausreichend ist, sollen besondere Entwicklungen bzw. besonderer Feature-Reichtum hervorgehoben werden durch Vergabe des maximalen Bewertungsfaktors. Auch die Komplexität der Verwendung der Komponenten aus der Sicht der Entwicklung soll hier einfließen. Berücksichtigt werden sollte hier noch, dass bei Realisierungsmöglichkeit einer Funktion durch mehrere Komponenten des Frameworks auch auf alle Varianten eingegangen werden soll, wobei die Bewertung sich stets nach der Komponente mit der höchsten Anforderungserfüllung richten soll.

In diesem Zusammenhang soll auch der Teilaspekt „Komponentenarchitektur“ beleuchtet werden, der mehr der Forschungsdisziplin Softwareengineering zuzuordnen ist. In diesem Kontext soll geprüft werden, in wie weit die Komponente nach moderner Softwarearchitektur (vgl. Boehm, 2006) entwickelt worden ist. Besonderes Augenmerk soll hier darauf gelegt werden, ob beispielsweise der JavaScript-Code der Komponente für weitere Komponenten wiederverwendet werden kann.

Im Anschluss daran soll der Aspekt „Funktionsweise bei nicht verfügbarem JavaScript“ untersucht werden, was bedeutet, dass zu prüfen ist, ob die entsprechende Komponente bei nicht verfügbarem JavaScript noch funktioniert und falls ja bis zu welchem Grad. Dieser Punkt wird auch innerhalb der BITV-Tests explizit angesprochen, soll aber aufgrund seiner zentralen Bedeutung für barrierefreie Webapplikationen explizit genannt und untersucht werden.

¹² SAGA = Standards und Architekturen für E-Government-Anwendungen.

¹³ BITV = Barrierefreie Informationstechnikverordnung; vgl. <http://www.gesetze-im-internet.de/bitv/BJNR265400002.html>.

Ferner ist die Erfüllung der Barrierefreiheitskriterien nach BITV zu prüfen, was anhand der Checklisten zur BITV und durch Tool-Unterstützung¹⁴ erfolgen kann.

Eine Prüfung des Barrierefreiheitsgrades gemäß der W3C-Konformitätsrichtlinien entfällt, da diese ihre Entsprechung bereits durch den BITV besitzen.

Neben dem Aspekt der Barrierefreiheit spielt jedoch auch noch die Anpassbarkeit eine nicht zu unterschätzende Rolle. So besteht aufgrund von Themen wie Corporate Identity (vgl. Debije-Meessen und Jansen, 2006) oder der zunehmenden Einbettung von Anwendungen in Portale (vgl. Duffner, 2003) die Anforderung, dass Komponenten hinsichtlich der Visualisierung flexibel anpassbar sind. Dieser Aspekt soll im Rahmen der Rich Client-Funktionalitätsbetrachtung analysiert werden.

Schließlich soll noch die Browserinteroperabilität jeder Komponente geprüft werden, um festzustellen, inwieweit ein Einsatz der Komponente und auch des zugehörigen Frameworks derzeit sinnvoll erscheint.

3.3 Abbildung der Detailanforderungen auf die Untersuchungsgegenstände

3.3.1 Beschreibung

In diesem Abschnitt sollen die verschiedenen Detailanforderungen spezifiziert und auf die zu prüfenden Steuerelemente abgebildet werden.

3.3.2 Rich-Client-Funktionalität

3.3.2.1 Dialoge

Im Schnittfeld der zu prüfenden Dialog-realisierenden Komponente und dem Prüfaspekt Rich-Client-Funktionalität ergeben sich die folgenden primären Teilanforderungen an Dialoge.

Request-Lebenszyklus-Persistenz

Darunter ist zu verstehen, dass eine Prüfung vorzunehmen ist, ob der Dialog nach einem Submit¹⁵ seinen vormaligen Anzeigezustand beibehält. Somit darf bei Anzeige eines Dialogs vor einem Submit dieser nicht nach dem Submit automatisch geschlossen werden und umgekehrt. Dabei handelt es sich um einen bedeutenden Aspekt, der der Anforderung Ausdruck verleiht, die Zustandslosigkeit des Web zu kompensieren (vgl. Brown, 2000), was wiederum die Grundlage für komplexe und zustandsbehaftete Webanwendungen bildet und mit Einführung der ViewState-Technik (vgl. Mitchell, 2004) im Rahmen von ASP.NET erstmalig weite Verbreitung fand.

Anpassbarkeit

Die Dialog-Komponente soll das Erscheinungsbild eines Dialogs ohne großen Entwicklungsaufwand zur Verfügung stellen, d.h. wesentliche Elemente wie die Titelleiste und zugehörige Buttons sollten unterstützt werden. Dennoch sollte der Dialog völlig flexibel konfigurierbar und frei anpassbar sein. Auch Theming-Mechanismen sind zu untersuchen.

¹⁴ Vgl. Truvex BITV Check - <http://checkwebsite.erigami.com/bitv.html>; ATRC Web Accessibility Checker unter <http://checker.atrc.utoronto.ca/servlet/ShowGuide?name=bitv-1-0-L2.xml>.

¹⁵ Der Vorgang des Abschickens der Daten eines Webformulars an einen Webserver.

Verhalten

Das Verhalten eines Dialogs einer Desktopapplikation soll auch in der Webimplementierung möglichst nah nachempfunden werden. Dazu zählen vor allem Aspekte wie Standardtastenkürzel (z.B. Escape für Auslösung der Abbrechen-Taste oder zum Schließen des Dialogs) und Unterstützung von Dialogmodalität (vgl. Ananiev und Nourie, 2006).

Entwicklungsunterstützung

Eine besondere Unterstützung in einer Entwicklungsumgebung, z.B. Eclipse¹⁶ 3.3 über Plugins oder ähnliche Technologien, soll in diesem Punkt hervorgehoben werden, da dieser Aspekt die Akzeptanz bei den Entwicklern und damit auch die Verbreitung des Produkts steigern kann und zudem wichtig im Hinblick auf den Aspekt Produktivität ist.

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt soll an mehreren Stellen, ohne Copy & Paste-Wiederverwendbarkeit, eingesetzt werden können.

Definierter Datenaustauschprozess

Es ist zu prüfen, ob eine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen, besteht.

Standard-Dialoge für MessageBoxen

Es sollte eine Reihe von Standard-Dialogen unterstützt werden, um komplexere Abfragen als die mit HTML-Bordmitteln realisierbaren confirm- und alert-Meldungen zu erstellen.

3.3.2.2 Registerkarten

Im Schnittfeld der zu prüfenden Registerkarten-realisierten Komponente und dem Prüfaspekt Rich-Client-Funktionalität ergeben sich die folgenden verfeinerten Teilanforderungen an Registerkartenkomponenten.

Request-Lebenszyklus-Persistenz

Darunter ist zu verstehen, dass eine Prüfung vorzunehmen ist, ob das Registerkartensteuerelement nach einem Submit¹⁷ seinen vormaligen Anzeigezustand beibehält. Somit muss die vor dem Submit angezeigte Registerkarte (Tab) nach dem Submit weiterhin dargestellt werden, ohne dass z.B. stets auf den ersten Tab zurückgesprungen wird. Wie in Kapitel 3.3.2.1 angesprochen, handelt es sich hierbei um einen bedeutenden Aspekt, der der Anforderung Ausdruck verleiht, die Zustandslosigkeit des Web zu umgehen.

Anpassbarkeit

Das Registerkartensteuerelement soll hinsichtlich der Tabknöpfe und Tabinhalte flexibel anpassbar sein. So sollen neben einfachen Textbeschriftungen auch Symbole angezeigt werden können. Der Registerkarteninhalt muss völlig frei belegbar sein. Auch Theming-Mechanismen können in diesem Rahmen betrachtet werden.

Verhalten

Das Verhalten muss sich dadurch auszeichnen, dass ein Anklicken eines Registerkartenknopfes den zugehörigen Registerkarteninhalt anzeigt und die restlichen Registerkarten des

¹⁶ Eine populäre Entwicklungsumgebung, primär mit Java-Fokus. Vgl. <http://www.eclipse.org/>.

¹⁷ Der Vorgang des Abschickens der Daten eines Webformulars an einen Webserver.

Registerkartensteuerelement verbirgt. Eine Ansteuerung der Tabknöpfe mit der Tastatur ist zu prüfen.

Entwicklungsunterstützung

Besondere GUI-Designer-Unterstützung, wie z.B. eine WYSIWYG¹⁸-Befüllung der Registerkarteninhalte, soll unter diesem Aspekt untersucht werden.

3.3.2.3 Menüs

Im Schnittfeld der zu prüfenden Menüsteuerelement-realisierenden Komponente und dem Prüfaspekt Rich-Client-Funktionalität ergeben sich die folgenden verfeinerten Teilanforderungen an Menüsteuerelemente.

Anpassbarkeit

Das Menüsteuerelement soll hinsichtlich der Darstellung angepasst werden können, was dadurch gegeben sein kann, dass beispielsweise neben reinen Textbeschriftungen auch Ikonen unterstützt werden. Auch Theming-Mechanismen sind zu untersuchen.

Verhalten

Eine komplette Ansteuerung des Menüs mit Tastatur sowie eine Anzeige der Untermenüpunkte bei Auswahl eines übergeordneten Menüs ist zu prüfen. Aufgrund der im Sinne von Barrierefreiheit empfohlenen Beschränkung auf möglichst flache Navigationsstrukturen wird es als ausreichend befunden, wenn die zu untersuchende Menükomponente lediglich mindestens zwei Gliederungsebenen zulässt. Auch die Unterstützung von Tastenkombinationen ist unter diesem Punkt abzu prüfen.

Entwicklungsunterstützung

Besondere GUI-Designer-Unterstützung, wie z.B. eine WYSIWYG-Befüllung der Registerkarteninhalte, soll unter diesem Aspekt untersucht werden.

3.3.2.4 Symbolleisten

Als nächster Punkt soll die Realisierung von Symbolleistenkomponenten näher betrachtet werden. Dazu ergeben sich die folgenden Detailanforderungen im Kontext des übergeordneten Aspekts Rich Client-Funktionalität.

Anpassbarkeit

Das Symbolleistensteuerelement soll hinsichtlich der Darstellung angepasst werden können, z.B. durch Unterstützung von Symbolen neben reinen Textbeschriftungen. Auch Theming-Mechanismen sind zu betrachten.

Verhalten

Eine komplette Ansteuerung des Menüs mit Tastatur wird als Vorgabe gegeben. Die Spezifikation von serverseitigen Ereignisbehandlungsroutinen ist zu unterstützen.

¹⁸ WYSIWYG = What you see is what you get; Prinzip, welches beschreibt, dass die Ausgabe von durch den Benutzer zusammengestellten Entitäten live während der Erstellung angezeigt wird, sodass der jeweilige Anwender bereits während der Ausübung der besagten Tätigkeit die Endresultate ersehen kann.

Entwicklungsunterstützung

Besondere GUI-Designer-Unterstützung, wie z.B. eine Anpassung der Symbolleisteigenschaften über Werkzeuge der Entwicklungsumgebung sind hier zu prüfen.

3.3.2.5 Hierarchische Steuerelemente (Tree Views)

Unter dem Aspekt der hierarchischen Steuerelemente soll geprüft werden, welche Realisierungen zur Visualisierung hierarchischer Daten zur Verfügung stehen. Als klassische Lösungen in der Desktopapplikationsentwicklung sind hier sogenannte Tree Views zu nennen, die über Expandierungs- und Kollabierungssymbole untergeordnete Elemente anzeigen bzw. verbergen. Folgende Anforderungen sollen dabei geprüft werden.

Request-Lebenszyklus-Persistenz

Mit dieser Anforderung soll erreicht werden, dass das jeweilige Steuerelement nach einem Submit eine unveränderte Darstellung aufweist, was bedeutet, dass der Expandierungs- bzw. Kollabierungszustand bewahrt werden soll. Wie in Kapitel 3.3.2.1 erörtert, ist dieser Aspekt auch hier als äußerst relevant zu bewerten, da damit die Problematik der Zustandslosigkeit des Web ausgeglichen werden soll.

Anpassbarkeit

Das TreeView-Steuerelement soll neben den Textbeschriftungen auch beliebige Elementicons sowie auch Kollabier- und Expandiericons unterstützen. Auch Theming-Mechanismen sollen im Rahmen des Aspekts Anpassbarkeit betrachtet werden.

Verhalten

Bei Anklicken eines Tree View-Knotens ist eine definierte serverseitige Aktion auszuführen. Bei Anklicken eines Kollabier- und Expandiericons ist hingegen ein Aus- bzw. Einklappen der untergeordneten Elemente vorzunehmen. Eine Bedienung komplett über die Tastatur wird als Pflichtanforderung vorgegeben.

Entwicklungsunterstützung

Aufgrund der theoretisch beliebig tiefen Verschachtelungen von Tree View-Komponenten und der daraus resultierenden, in Abhängigkeit von den zugeordneten Daten potentiell auftretenden Unübersichtlichkeit im jeweiligen Designwerkzeug der entsprechenden Entwicklungsumgebung, werden keine besonderen Anforderungen an den Aspekt Entwicklungsunterstützung gestellt.

3.3.2.6 Fortschrittsanzeige

Das jeweilige Fortschrittsanzeige-Steuerelement soll Auskunft über den Fortschritt einer andauernden Aktion durch graphische Visualisierung bieten. Folgende Anforderungen sind gegen diese Komponente zu prüfen

Anpassbarkeit

Das Fortschrittsanzeige-Steuerlement soll hinsichtlich der Darstellung frei anpassbar sein. Dies schließt die Grafik zur Visualisierung des Fortschritts mit ein. Auch Theming-Mechanismen sind zu untersuchen.

Verhalten

Während eines länger andauernden Vorgangs ist eine Fortschrittsanzeige einzublenden. Diese kann eine inkrementelle Visualisierung darstellen, oder alternativ einfach eine Animation abspielen, um dem Anwender mitzuteilen, dass der aktuelle Vorgänge gegenwärtig noch läuft.

Entwicklungsunterstützung

Es soll eine freie Festlegung der Fortschrittsanzeige-Inhalte über die jeweilige Entwicklungsumgebung graphisch unterstützt werden.

3.3.2.7 Infotips

Bei Infotip-Komponenten handelt es sich um informative Popups, die bestimmte Komponenten, über welchen sich der Mauszeiger befindet, ausführlicher als ein herkömmlicher Tooltip beschreiben (vgl. Microsoft Corporation, 2007a).

Anpassbarkeit

Die Infotip -Komponente soll hinsichtlich des anzuzeigenden Inhaltes völlig frei anpassbar sein. Zudem ist eine mehrzeilige Anzeige und die Visualisierung von Icons vorzusehen. Ferner sollen Standardansichten unterstützt werden, um häufige Anforderungen ohne großen Aufwand abzudecken. Auch Theming-Mechanismen sind zu prüfen.

Verhalten

Ein Infotip ist bei Positionierung des Mauszeigers über der zu erläuternden Komponente einzublenden und beim Verlassen der zu erläuternden Komponente wieder auszublenden.

Entwicklungsunterstützung

Es soll eine freie Befüllung der Infotip -Komponente über die jeweilige Entwicklungsumgebung graphisch unterstützt werden.

3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Dieser Aspekt ist bei allen spezifizierten Komponenten, ohne spezifische Unterschiede, zu prüfen. Aufgrund Anforderungen der Barrierefreiheit und der allgemeinen Forderung nach Unabhängigkeit des Funktionierens einer Webanwendung vom Endgerät und Browser des Anwenders ist dieser Aspekt als besonders wichtig zu bewerten.

3.3.4 Barrierefreiheit nach BITV¹⁹

Für die Prüfung der BITV-Kriterien nach den Prioritäten 1 und 2 soll auf die offizielle Checklisten²⁰ verwiesen werden. Jacobs und Kruse bieten eine kommentierte Fassung der Checklistenanforderungen an (vgl. Jacobs, 2005). Im Rahmen der Untersuchung sollen die 14 daraus hervorgehenden Anforderungen aus Priorität 1 und 2 stets in einem Punkt behandelt werden, sodass daraus 14 Prüfaspekte resultieren.

3.3.5 Browserinteroperabilität

Es ist sicherzustellen, dass die jeweilige Komponente unter allen gängigen Browsern voll funktionsfähig ist. Im Detail sollen die Browser

- Internet Explorer 6.0+,
- Mozilla Firefox 2.0+,
- Opera 9+

¹⁹ BITV = Barrierefreie Informationstechnikverordnung; vgl. <http://www.gesetze-im-internet.de/bitv/BJNR265400002.html>.

²⁰ Checkliste zur Priorität 1 unter <http://www.wob11.de/checklistbitvprio1.html>; Checkliste zur Priorität 2 unter <http://www.wob11.de/checklistbitvprio2.html>.

- und Safari 3.1+ unterstützt werden.

3.4 Rahmenparameter

Trotz der primär objektiven Untersuchung der im nächsten Kapitel vorgestellten Frameworks hinsichtlich der in diesem Abschnitt formulierten Anforderungen, sind subjektive Einflüsse nicht komplett auszuschließen, da Frameworks von komplett verschiedenen Plattformen mit teils plattformspezifischen Funktionalitäten verglichen werden. Diese „Störfaktoren“ sind jedoch in ihrer Gesamtwirkung als unwesentlich zu bewerten, sodass keine Verzerrung des Gesamtergebnisses zu erwarten ist.

4 Produktauswahl und Bewertung

4.1 Überblick

Im Rahmen der Untersuchung sollen folgende reichhaltige Webapplikations-Komponentenframeworks betrachtet werden.

- ASP .NET-basierte Lösungen
 - ASP.NET 3.5 (Standard)
 - AJAX Control Toolkit²¹
 - ComponentArt Web.UI²²
 - NetAdvantage for ASP. NET²³
 - Telerik RadControls for ASP.NET AJAX²⁴
 - Developer Express ASPxperience Suite²⁵
- JSF-basierte Lösungen
 - JSF (Standard)
 - MyFaces Tomahawk²⁶
 - JBoss RichFaces²⁷
 - Oracle ADF Rich Client²⁸
 - NetAdvantage for JSF²⁹

Die angeführten Webapplikationsframeworks stellen dabei vonseiten der Popularität, dem Verbreitungsgrad und der Zukunftsträchtigkeit die bedeutendsten Untersuchungsgegenstände dar und sind im Rahmen von eGovernment in Bayern relevant, da im bayerischen eGovernment-Umfeld webbasierte Lösungen hauptsächlich im Java- und ASP.NET-Umfeld angesiedelt sind (vgl. Bayerisches Landesamt für Finanzen, 2008).

Zu beachten ist, dass Frameworks, welche zusätzliche Plugins im Browser des Anwenders erfordern, von vornherein bei der Untersuchung ausgeschlossen werden, da die Zielsetzung der Untersuchung ist, Frameworks mit einem Maximum an Barrierefreiheit ausfindig zu machen, welche somit keine besonderen Anforderungen an die Browser der Nutzer stellen. Damit scheiden Lösungen wie Adobe Flex (vgl. Adobe Systems Incorporated, 2006) oder Silverlight (vgl. Rader et al., 2007) im Rahmen dieser Untersuchung aus.

Darüber hinaus sind Frameworks, für welche noch keine breite Browserunterstützung vorliegt oder in naher Zukunft verfügbar sein wird, im Rahmen dieser Untersuchung ebenfalls ausgeschlossen. Hierzu zählen auch komplett neue Ansätze wie XForms (vgl. Dubinko, 2003).

Das vielversprechende und innovative Projekt Visual WebGUI³⁰, welches die herkömmliche und komfortable Entwicklung von Desktopanwendungen und die Übersetzung in Webanwendungen gestattet, wird von der Untersuchung ebenfalls ausgeschlossen, da dieses komplett auf AJAX und

²¹ Vgl. <http://www.asp.net/ajax/ajaxcontroltoolkit/samples/Default.aspx>.

²² Vgl. <http://www.componentart.com/webui/>.

²³ Vgl. <http://www.infragistics.com/dotnet/netadvantage/aspnet.aspx#Overview>.

²⁴ Vgl. <http://www.telerik.com/products/aspnet-ajax/overview.aspx>.

²⁵ Vgl. <http://www.devexpress.com/Products/NET/WebForms/>.

²⁶ Vgl. <http://myfaces.apache.org/tomahawk/index.html>.

²⁷ Vgl. <http://www.jboss.org/jbossrichfaces/>.

²⁸ Vgl. <http://www.oracle.com/technology/products/adf/adffaces/index.html>.

²⁹ Vgl. <http://www.infragistics.com/java/netadvantage/jsf.aspx#Overview>.

³⁰ Vgl. <http://www.visualwebgui.com/>.

JavaScript setzt und damit bei Nicht-Verfügbarkeit von JavaScript nicht einsetzbar ist, aber aus Gründen der Barrierefreiheit keine Abhängigkeit von JavaScript vorliegen darf.

4.2 Bewertung

Im Hinblick auf die Bewertung soll für minimale Anforderungserfüllung ein Erfüllungswert von 0,0, für teilweise vorliegende Anforderungserfüllung ein Wert von 0,5 und für fortgeschrittene Erfüllung ein Wert von 1,0 vergeben werden. Zusätzlich wird pro Teilanforderung ein Gewichtungsfaktor festgelegt, mit welchem die jeweilige Teilanforderung bei der Summenbildung multipliziert wird.

Der Grad der Gesamtanforderungserfüllung g_j des Frameworks j errechnet sich dabei dann aus dem gewichteten, arithmetischen Mittel aller Teilanforderungserfüllungswerte (vgl. Abbildung 1).

$$g_j = \frac{\sum (r_i * a_{(j,i)})}{\sum (r_i)}$$

g_j = Grad der Gesamtanforderungserfüllung des Frameworks j

r_i = Relevanzfaktor der Anforderung i

$a_{(j,i)}$ = Erfüllungsgrad der Anforderung i bei Framework j

Abbildung 1: Definition des Grades der Gesamtanforderungserfüllung

Für die Bewertung ist zu berücksichtigen, dass die alleinige Realisierbarkeit einer Funktionalität keine gute Bewertung ausmacht. Lösungen, welche Zweckentfremdungen oder Workarounds erfordern, sollen aufgrund der Anforderungen aus dem Bereich modernes Software Engineering damit schlechter bewertet werden, als vom jeweiligen Framework vorgesehene und offiziell unterstützte Lösungswege. Falls Anforderungen in einem bestimmten Kontext nicht anwendbar sind, z.B. barrierefreie Bilddarstellung bei Dialogkomponente, welche keine Bilder unterstützt, wird die maximale Leistungszahl vergeben, da keinerlei Verstoß gegen die entsprechende Barrierefreiheitsanforderung vorliegt.

Auch resultieren die Bewertungen aufgrund der agilen und experimentellen Fragestellung in nahezu allen Fällen auf eigenen Tests anstelle von Literaturarbeit, was auch daran liegt, dass zu vielen der Frameworks noch keine oder nur eine sehr unvollständige Dokumentation vorliegt. Bei Verfügbarkeit entsprechender, zielführender und passender Literatur wird jedoch zusätzlich auf diese verwiesen.

4.3 Gewichtungsfaktoren

Um die Relevanz der zu prüfenden Anforderungen in der Untersuchung widerzuspiegeln, wird jedem Anforderungspaket ein Gewichtungsfaktor zugewiesen:

- Rich-Client-Funktionalität: 1,0
- Funktionsweise bei nicht-verfügbarem JavaScript: 1,0
- Barrierefreiheit nach BITV: 1,0
- Browserinteroperabilität: 1,0

Außerhalb der Berechnung der Anforderungserfüllung des jeweiligen Frameworks soll die Funktionsweise bei nicht-verfügbarem JavaScript zusätzlich betrachtet werden, da es sich hierbei

nicht nur um eine hinreichende, sondern eine notwendige Anforderung handelt, d.h. falls diese Anforderung nicht erfüllt ist, scheidet das Framework hinsichtlich der besagten Komponente komplett aus, da dies einen zentralen Verstoß gegen den Aspekt Barrierefreiheit darstellt.

5 Evaluierung

5.1 Präambel

5.1.1 Basisplattformen- und Frameworks

Wie bereits unter Kapitel 4.1 erläutert, sollen primär auf ASP .NET (mit Ajax-Erweiterungen) und JSF basierende Webapplikationsframeworks mit Rich Client Frameworks untersucht werden, wobei eine Beschränkung auf die bekanntesten und verbreitetsten, aber auch innovativsten Lösungen vorgenommen werden soll.

ASP .NET³¹ (vgl. Esposito, 2004; Esposito, 2005) ist ein auf dem .NET-Framework basierendes, komponentenorientiertes Webentwicklungs-Framework zur Realisierung von dynamischen Webanwendungen und Webdiensten von Microsoft.

Bei ASP .NET AJAX (vgl. McClure et al., 2006), welches vor offizieller Veröffentlichung als Atlas bezeichnet worden ist, handelt es sich nach Wenz (2007: 3) um eine Sammlung neuer Microsoft-Technologien, die es Web- und insbesondere ASP.NET 2.0-Entwicklern, ermöglichen, vereinfacht Webseiten mit Ajax-Unterstützung zu realisieren.

Bei JSF handelt es sich hingegen (vgl. Geary und Horstmann, 2007) um eine komponentenorientierte, auf der Java EE-Plattform³² basierende, Spezifikation von Sun zur Webentwicklung. Neben der Referenzimplementierung von Sun sind weitere freie (z.B. MyFaces) und kommerzielle (z.B. Oracle ADF Rich Client) Implementierungen erhältlich.

Im Folgenden sollen nun die auf den oben genannten Frameworks aufbauenden und in Kapitel 4.1 spezifizierten Rich Client-Komponentenframeworks für den Einsatzbereich Webentwicklung näher betrachtet werden, wobei angemerkt werden soll, dass aufgrund der unterschiedlichen JSF-Komponentennamen und Tagnamen im Hinblick auf die Groß- bzw. Kleinschreibung diese abwechselnd eingesetzt wird, was jedoch keinen semantischen Unterschied anzeigt.

Für einen weiteren, jedoch wesentlich kürzeren und auf anderen Anforderungen basierend durchgeführten AJAX-Frameworkvergleich wird auf die "JSF AJAX Component Library Feature Matrix"³³ verwiesen.

5.1.2 Evaluierte Produktversionen

Gegenstand dieser Untersuchung sind die jeweils zum Zeitpunkt der Evaluierung aktuellen Produktversionen, wenngleich einzelne Ausnahmen möglich sind.

Dabei ist zu berücksichtigen, dass im Rahmen der Produktevaluierung auf ältere Versionen zurückgegriffen werden kann, falls entsprechende Literatur für die aktuelle Version nicht im ausreichenden Maß zur Verfügung steht und zugleich keine funktionalen Unterschiede zwischen alter und neuer Version im Hinblick auf den zu prüfenden Aspekt vorliegen.

³¹ Vgl. <http://msdn2.microsoft.com/en-us/library/aa479045.aspx>.

³² Vgl. <http://java.sun.com/javaee/reference/>.

³³ Vgl. <http://www.ajaxjsf.net/>.

5.2 ASP.NET 3.5 (Standard)

5.2.1 Beschreibung

Bevor die Erweiterungsframeworks von ASP.NET analysiert werden, soll zunächst geprüft werden, in wie weit ASP.NET selbst die gestellten Anforderungen erfüllt.

5.2.2 Evaluierte Produktversion

Grundlage dieser Untersuchung ist Version 3.5 von ASP.NET.

5.2.3 Detailuntersuchung

5.2.3.1 Dialoge

Es werden keine Dialogsteuerelemente unterstützt.

Bewertung: 0,0

5.2.3.2 Registerkarten

Es werden keine Registerkartensteuerelemente unterstützt, wenngleich durch Kombination einfacher Steuerelemente wie des MultiView ein ähnliches Verhalten nachgebildet werden könnte.

Bewertung: 0,0

5.2.3.3 Menüs

5.2.3.3.1 Beschreibung

Menüs können über das Menu-Steuerelement (vgl. Abbildung 2) realisiert werden. Dieses dient primär der Anzeige von Navigationsstrukturen, kann aber auch anderweitig eingesetzt werden.

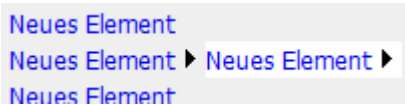


Abbildung 2: Das ASP.NET-Menu-Steuerelement

5.2.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Das Menu-Steuerelement ist flexibel anpassbar. So können globale Style-Klassen festgelegt werden, oder aber auch die Darstellung für jedes einzelne Menü-Element angepasst werden. Darüber hinaus ist es möglich, einem Menüelement neben Text auch Symbole hinzuzufügen. Auch eine Anpassung der Visualisierung des Selektionszustands wird unterstützt.

Bewertung: 1,0

Verhalten

Ein Anklicken eines TreeView-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses. Unterelemente werden in der Regel bei Bewegung des Mauszeigers über das entsprechende Menüelement angezeigt; jedoch nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Der Aufbau der Menüstrukturen wird durch den sogenannten Menüelement-Editor auf einfache Art und Weise ermöglicht.

Bewertung: 1,0

5.2.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Steurelement seine Funktionalität ein. Es werden keine Untermenüpunkte mehr angezeigt.

Bewertung: 0,0

5.2.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Menüsymbolen können Alternativtexte spezifiziert werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keine negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global diverse graphische Einstellungen vorgenommen werden können, inklusive auf die verschiedenen Elemente anzuwendende Vorder- und Hintergrundfarben. Es kann jedoch einem einzelnen Element nicht auf einfache Art und Weise eine spezifische CSS-Klasse zugewiesen werden.

Bewertung: 0,5

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet eine Zweckentfremdung des table-Elements bei dieser Komponente statt. Über Zusatzpakete wie die CSS Friendly Control Adapters³⁴ sind CSS-konforme Ausgaben in Form von ul- und li-Elementen möglich.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Jedoch können Überlappungen zwischen Menüinhalten und Inhalten der HTML-Seite auftreten, sodass nur noch eine schlechte Lesbarkeit der Inhalte möglich ist. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Untermenüs beim Bewegen des Mauszeigers auf ein Hauptmenü mehr angezeigt werden können.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist möglich.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen. Durch die CSS

³⁴ Vgl. <http://www.codeplex.com/cssfriendly>.

Friendly Control Adapters wäre jedoch eine Ausgabe in Form von strukturierten ul- und li-Tags denkbar.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung der Elemente kann über die CSS Friendly Control Adapters in Form von ul- und li-Tags erfolgen, was jedoch nicht als Standardimplementierung vorliegt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.2.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Die derzeitige Implementierung verursacht in verschiedenen Testszenarien Probleme beim Einblenden der untergeordneten Menüelemente.

Bewertung: 0,5

5.2.3.4 Symbolleisten

Es wird kein Steuerelement für den Anwendungsfall Symbolleiste im Rahmen des AJAX Control Toolkit ausgeliefert, weshalb alle Teilaspekte mit 0,0 zu bewerten sind.

Bewertung: 0,0

5.2.3.5 Hierarchische Steuerelemente (Tree Views)

5.2.3.5.1 Beschreibung

Hierarchische Strukturen können über das TreeView-Steuerelement (vgl. Abbildung 3) realisiert werden. Dieses dient primär der Anzeige von Navigationsstrukturen, kann aber auch anderweitig eingesetzt werden.



Abbildung 3: Das ASP.NET-TreeView-Steuerelement

5.2.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist.

Bewertung: 0,5

Anpassbarkeit

Das TreeView-Steuerelement ist flexibel anpassbar. So können Style-Klassen global festgelegt werden, oder aber auch die Darstellung für jedes einzelne Baumknoten-Element angepasst werden. Darüber hinaus ist es möglich, einem TreeView-Node (Baumknoten) neben Text auch Symbole hinzuzufügen. Auch eine Anpassung im Hinblick auf die Anzeige von Checkboxes in Baumknoten wird unterstützt. Es kann jedoch einem einzelnen Baumknoten keine CSS-Klasse auf einfache Art und Weise deklarativ zugewiesen werden.

Bewertung: 0,5

Verhalten

Ein Anklicken eines TreeView-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses, sofern JavaScript aktiviert ist. Unterelemente werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch ebenso nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Der Aufbau der Baumstrukturen wird durch den sogenannten TreeView-Knoten-Editor auf einfache Art und Weise ermöglicht.

Bewertung: 1,0

5.2.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität ein. Es werden keine untergeordneten Einträge mehr angezeigt.

Bewertung: 0,0

5.2.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können Alternativtexte spezifiziert werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global diverse graphische Einstellungen vorgenommen werden können. Auf Baumknotenebene kann jedoch keine CSS-Klasse komfortabel zugewiesen werden.

Bewertung: 0,5

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet eine Zweckentfremdung des table-Elements bei dieser Komponente statt. Über Zusatzpakete wie die CSS Friendly Control Adapters³⁵ sind aber CSS-konforme Ausgaben in Form von ul- und li-Elementen möglich.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Jedoch können Überlappungen zwischen Bauminhalten und Inhalten der HTML-Seite auftreten, sodass nur noch

³⁵ Vgl. <http://www.codeplex.com/cssfriendly>.

eine schlechte Lesbarkeit der Inhalte möglich ist. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass kein Auf- oder Zuklappen von untergeordneten TreeView-Knoten mehr funktioniert.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist möglich.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen. Durch die CSS Friendly Control Adapters wäre jedoch eine Ausgabe in Form von strukturierten ul- und li-Tags denkbar.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung der Elemente kann über die CSS Friendly Control Adapters in Form von ul- und li-Tags erfolgen, was jedoch nicht als Standardimplementierung vorliegt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.2.3.5.5 Browserinteroperabilität

Das TreeView-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.2.3.6 Fortschrittsanzeige

5.2.3.6.1 Beschreibung

Länger andauernde Vorgänge können über Fortschrittsanzeige-Steuerelemente visualisiert werden. Wenngleich mit dem UpdateProgress-Steuerelement (vgl. Abbildung 4) keine Information über den prozentualen Fortschritt eines Vorgangs angeboten werden kann, ist es möglich, eine Animation einzublenden, die den Benutzer darüber informiert, dass der Prozess noch am Laufen ist. Der anzuzeigende Inhalt lässt sich dabei völlig frei konfigurieren.

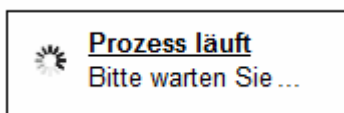


Abbildung 4: Das ASP.NET-UpdateProgress-Steuerelement

5.2.3.6.2 Rich-Client-Funktionalität

Anpassbarkeit

Das UpdateProgress-Steuerelement ist flexibel anpassbar. So kann der komplette anzuzeigende Inhalt, inklusive Animationen, völlig frei unter einem sogenannten ProgressTemplate vorgegeben werden.

Bewertung: 1,0

Verhalten

Bei Ausführung einer beliebigen AJAX-Anforderung (über das Update Panel-Steuerelement) wird automatisch das UpdateProgress-Steuerelement aktiv, welches den definierten Inhalt während der Laufzeit des Vorgangs einblendet. Das Steuerelement wird jedoch naturgemäß nur bei AJAX-Requests tätig.

Bewertung: 1,0

Entwicklungsunterstützung

Der Inhalt des Steuerelements kann sowohl über die Codeansicht als auch über den Designer vorgegeben werden.

Bewertung: 1,0

5.2.3.6.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt die Komponente ihre Funktionalität komplett ein.

Bewertung: 0,0

5.2.3.6.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können Alternativtexte spezifiziert werden. Dies liegt aber aufgrund der freien Anpassbarkeit vollständig in der Hand des Entwicklers.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass aufgrund der freien Anpassbarkeit der anzuzeigenden Inhalte durch den Entwickler, keine Bewertungsgrundlage hinsichtlich dieses Aspekts vorliegt, sodass wie beim vorherigen Punkt die maximale Leistungsstufe ohne Abzug vergeben wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die anzuzeigenden Inhalte werden komplett durch den Entwickler vorgegeben. Lediglich der Rahmen, wobei es sich hier um einen div-Bereich handelt, wird durch die Komponente vorgegeben.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin ohne Einschränkung benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente ihre Funktion komplett einstellt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen. Jedoch kann die Visualisierung des Prozessfortschritts durch diese Komponente bereits als Unterstützung der Zeitvisualisierung betrachtet werden.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ergibt im Hinblick auf die Intention dieser Komponente keinen Sinn.

Bewertung: 1,0

Anforderung 10

Diese Komponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames eingesetzt. Die Strukturierung von Daten des Steuerelements obliegt aufgrund der freien Anpassbarkeit dem Entwickler.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen, wie z.B. die Bedeutung der Fortschrittskomponente, werden nicht automatisch bereitgestellt. Eine sinnvolle Gruppierung der beinhaltenden Inhalte obliegt dem Entwickler.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.2.3.6.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.2.3.7 Infotips

5.2.3.7.1 Beschreibung

Es wird keine Komponente zur Realisierung eines Infotips mit ausgeliefert, wenngleich durch Kombination verschiedener bestehender Steuerelemente und Programmieraufwand eine Komponente mit ähnlicher Funktionalität geschaffen werden kann.

Bewertung: 0,0

5.2.4 Fazit

ASP.NET bildet eine umfangreiche und anpassungsfähige Basis zur Erstellung von reichhaltigen Webanwendungen. Für Rich Client-Webapplikationen fehlen jedoch noch viele Standard-Komponenten, wie Dialoge und Registerkartenkomponenten. Dieser Umstand unterstreicht die architekturelle Strategie ASP.NET als Basisplattform zu verwenden und auf dessen Grundlage Rich Client-Frameworks zu nutzen, um die offenen Lücken zu schließen. Aus diesem Grund werden im Folgenden darauf basierende Frameworks näher analysiert.

5.3 AJAX Control Toolkit

5.3.1 Beschreibung

Beim Ajax Control Toolkit handelt es sich nach Wenz (2007: 6) um eine Sammlung serverseitiger Komponenten, die beeindruckende Ajax-Funktionalitäten bei minimalem Entwicklungsaufwand bereitstellen. Das Ajax Control Toolkit ist ein Open Source-Projekt, wenngleich Microsoft das Projekt koordiniert und steuert, um die Qualität sicherzustellen. Es gibt jedoch keinen offiziellen Support Microsofts dafür. Nach Wenz (2007: 177) erweitern die Komponenten des AJAX Control Toolkits Microsofts ASP .NET AJAX.

Zu beachten ist, dass eine Großzahl der Komponenten nach dem Extender-Pattern (vgl. Esposito, 2008) realisiert worden ist, sodass beispielsweise die Komponente PopupControl durch einen sogenannten PopupControlExtender verwirklicht wird. Als Namenskonvention wird dabei stets die Komponente Xxx durch den Extender XxxExtender realisiert, wobei ein Extender allgemein eine bestehende Komponente um zusätzliche Funktionalität erweitert.

5.3.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 3.5.20229 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.3.3 Detailuntersuchung

5.3.3.1 Dialoge

5.3.3.1.1 Beschreibung

ASP .NET Ajax Control Toolkit unterstützt die Realisierung von Dialogen durch drei verschiedene Komponenten.

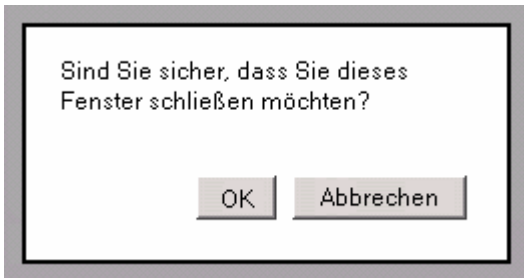


Abbildung 5: Das AJAX Control Toolkit-ConfirmButton-Control

Das PopupControl (vgl. Abbildung 7) erlaubt die Anzeige nicht-modaler Bereiche, die über entsprechendes Theming als Dialoge dargestellt werden können.

Die ModalPopup-Komponente (vgl. Abbildung 6) auf der anderen Seite gestattet die Erstellung modaler Dialoge mit Dialog-Aussehen.

ConfirmButton (vgl. Abbildung 5) hingegen ermöglicht die Darstellung einer MessageBox, einer sogenannten ConfirmBox, mit den Auswahlmöglichkeiten OK und Abbrechen.

Angemerkt werden soll, dass falls bei der folgenden Untersuchung von „die Komponente“ gesprochen wird, alle drei der oben genannten Steuerelemente gemeint sind, was im Fall eines gleichen Verhaltens oder einer gleichen Charakteristik aus Gründen der Vereinfachung durchgeführt werden kann.

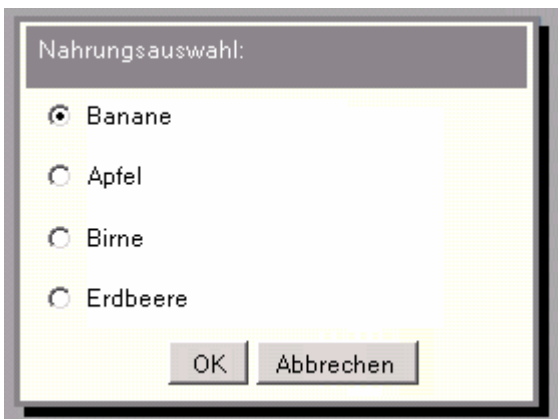


Abbildung 6: Das AJAX Control Toolkit-ModalPopup-Control

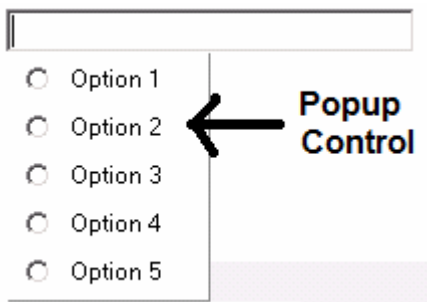


Abbildung 7: Das AJAX Control Toolkit-PopupControl-Steuerelement

5.3.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Komponenten PopupControl, ModalPopup und ConfirmButton wird nicht über Requestgrenzen hinweg persistiert.

Bewertung: 0,0

Anpassbarkeit

Der Dialog-Inhalt kann bei Verwendung von ModalPopup völlig frei angepasst werden. Ebenso könnte eine eigene Titelleiste frei vorgegeben werden. Dasselbe trifft für die Komponente PopupControl zu. ConfirmButton erlaubt hingegen nur sehr eingeschränkt die Vorgabe des anzuzeigenden, die jeweilige Frage beinhaltenden, Textes.

Da Dialoge sich jedoch in der Regel an bestimmten formalen Kriterien orientieren, was die Visualisierung einer Titelleiste mit z.B. einem Schließen-Button vorsieht, soll auch dieser Aspekt in die Gesamtbewertung des Punktes Anpassbarkeit miteinfließen. Keine der Komponenten ModalPopup, PopupControl und ConfirmButton bietet standardmäßig eine Titelleiste oder Standardschaltflächen an. Dies ist für jeden darauf basierenden Dialog durch den jeweiligen Entwickler komplett neu zu implementieren.

Bewertung: 0,5

Verhalten

Bei jeder der genannten drei Komponenten, kann die auslösende Komponente zur Anzeige des Dialogs frei konfiguriert werden. Es gibt jedoch keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Auch Standardtastenkürzel werden nicht ohne Entwicklungsaufwand unterstützt.

Bewertung: 0,0

Entwicklungsunterstützung

Eine gute Unterstützung liegt bei Benutzung von Visual Studio 2008 vor. Das ModalPopup-Steuerelement kann dabei über den Visual Studio-Standardeditor frei mit zu beinhaltenden Komponenten befüllt werden; jedoch nur, falls die Sichtbarkeit des Dialogbereichs auf sichtbar gestellt wird (Style-Attribut muss display-Wert Leerstring, block oder inline besitzen). Zur Laufzeit muss dieser jedoch initial auf unsichtbar gestellt werden. Eine explizite Unterstützung über spezifische Designerelemente liegt jedoch nicht vor.

Bewertung: 0,5

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt eines Dialoges kann nicht ohne Verwendung weiterer benutzerdefinierter Komponenten bzw. eigenem Entwicklungsaufwand an mehreren Stellen, ohne Copy & Paste-Wiederverwendbarkeit, eingesetzt werden können, da im Rahmen der Dialog-Implementierung der genannten Komponenten ein expliziter Mechanismus für diese Problemstellung definiert worden ist.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es sind lediglich Confirm-MessageBoxen durch die Komponente ConfirmButton vorgesehen, wobei eine sehr eingeschränkte Umsetzung vorliegt, da die anzuzeigenden Buttons keine Anpassung zulassen und auch keine anzuzeigenden Standardsymbole unterstützt werden.

Bewertung: 0,0

5.3.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellen unter Kapitel 5.3.3.1 beschriebenen Steuerelemente ihre Funktion ein. So ist keine Anzeige und auch kein Schließen eines Dialogs mehr möglich.

Bewertung: 0,0

5.3.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass keine spezielle CSS-Unterstützung vorliegt, besonders bei Betrachtung der Komponente ConfirmButton. Hier kann nur über den Umweg der Spezifikation eines PopupControl maximale Flexibilität erreicht werden, sodass sämtliche anzuzeigende Komponenten über die ASP.NET-Standardmechanismen visuell angepasst werden können.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt.

Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Keine der Dialog-Komponenten basiert auf Tabellen, weshalb keine Anwendbarkeit dieses Prüfaspekts vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung grundsätzlich noch verwendbar, jedoch schwerer bedienbar, da kein Unterschied mehr zwischen angezeigten und verborgenen Dialogen existiert. Zudem treten in der aktuellen Konfiguration bei manchen Browsern Seiteneffekte auf, die sich dadurch äußern, dass bei mehrmaligem Ein- und Ausblenden von Dialogen ein zunehmend größerer Leerraum sich auf der jeweiligen Webseite manifestiert. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (Anzeigen, Verbergen) existiert jedoch keine explizite Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Komponenten um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich dieses Steuerelemente keine sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden jedoch auch keine spezifischen Elemente, wie z.B. Tittleisten, vorab mitausgeliefert, was als Defizit bewertet werden kann.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.3.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.3.3.2 Registerkarten

5.3.3.2.1 Beschreibung

ASP .NET Ajax Control Toolkit unterstützt die Realisierung von Registerkarten durch das Tabs-Steuerelement (vgl. Abbildung 8).

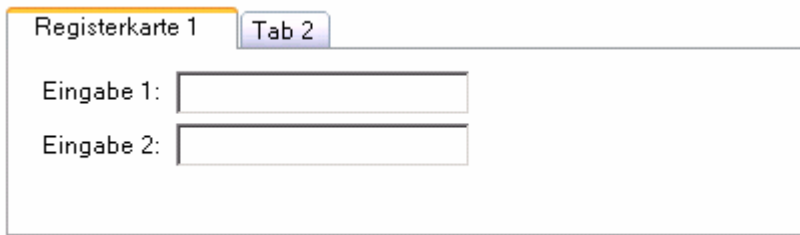


Abbildung 8: Das AJAX Control Toolkit-Tab-Control

5.3.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Tab-Komponente wird über Request-Grenzen hinweg persistiert, jedoch nur falls JavaScript aktiviert ist.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe sind jedoch nicht ohne Weiteres, z.B. um Symbole anpassbar.

Bewertung: 0,5

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist mit Ausnahme von Accesskeys bei deaktiviertem JavaScript nur äußerst eingeschränkt und nicht browserübergreifend möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Eine gute Unterstützung liegt bei Benutzung von Visual Studio 2008 vor. Registerkarten eines Registerkartensteuerelements können via Drag & Drop befüllt werden und die Registerkarten allgemein über Smart Tags verwaltet werden.

Bewertung: 1,0

5.3.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Tab-Control seine Funktionalität ein. Es ist kein Tabwechsel mehr möglich, da die Komponente nicht einmal mehr angezeigt wird.

Bewertung: 0,0

5.3.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die

ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass sowohl für die Komponente in ihrer Gesamtheit als auch für jeden einzelnen Tab die CSS-Klasse festgelegt werden kann. Die Inhalt einer Registerkarte können zudem völlig frei vorgegeben werden, was auch für die zugehörige CSS-Klasse gilt.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Tab-Komponenten basieren auf div-Elementen und nicht auf Tabellen, weshalb keine Anwendbarkeit dieses Prüfaspekts vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Alle Registerkarteninhalte werden aber ständig und untereinander dargestellt. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge innerhalb einer Registerkarte ist standardmäßig gegeben. Die Registerkartenknöpfe können jedoch in den meisten Browsern nicht fokussiert werden. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden jedoch auch keine spezifischen Elemente, wie z.B. Submit-Buttons oder Überschriften als Registerkartenknöpfe verwendet, was als Defizit zu bewerten ist.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.3.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.3.3.3 Menüs

5.3.3.3.1 Beschreibung

Menüs können unter dem AJAX Control Toolkit über die HoverMenu-Komponente (vgl. Abbildung 9) realisiert werden. Diese stellt aber nur die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element einen anderen frei definierbaren Bereich einzublenden, auf welchem z.B. auch Hyperlinks positioniert werden können, weshalb man hier nicht von „echten“ Menüs sprechen kann.

	Datensatz 1	45,00
Edit	Datensatz 2	12,23
Delete	Datensatz 3	19,14
	Datensatz 4	79,80
	Datensatz 5	14,25
	Datensatz 6	81,12

Abbildung 9: Das AJAX Control Toolkit-HoverMenu-Control

Daneben besteht noch die Möglichkeit SharePoint-ähnliche Menüs über das DropDown-Control (vgl. Abbildung 10) zu verwirklichen. Es handelt sich jedoch auch hier um keine echte Menü-Unterstützung, da die Menüeinträge über reguläre Links zu definieren und visualisieren sind.



Abbildung 10: Das AJAX Control Toolkit-DropDown-Control

5.3.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Da es sich bei beiden Komponenten um keine vollwertige Menüimplementierung handelt, können auch keine Menüelemente als solche oder verschachtelte Strukturen aufgebaut werden. Es wird

lediglich ein Container-Element eingeblendet, welches mit beliebigen Elementen füllbar ist, weshalb ein Maximum an Anpassbarkeit vorliegt. Jedoch bedeutet dies auch, dass keine vorkonfigurierten Profile vorliegen, um Desktopapplikationsmenü-ähnliche Menüs zu erstellen. Auch Icons und Accesskeys sowie Menüselektionseffekte müssen komplett neu implementiert werden, ohne Hilfe durch diese Komponente. Der Hover-Effekt des menüanzeigeauslösenden Elements ist hinsichtlich der Gestaltung anpassbar.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine reine Bedienung mit der Tastatur ist browserabhängig nicht oder nur schwer möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Da es sich nicht um echte Menükomponenten handelt, die eine Modellierung von Menüelementen erlauben, kann für die Menü-Modellierung auch nur der ASP.NET-Standarddesigner verwendet werden.

Bewertung: 0,5

5.3.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr angezeigt.

Bewertung: 0,0

5.3.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die Komponenten selbst nur die Anzeige eines Containersteuerelements bei Positionierung des Mauszeigers über einem zu erweiternden Steuerelement implementiert, sodass sämtliche sichtbaren GUI-Elemente in diesem Zusammenhang durch ihre eigenen, originär vorhandenen Style-Attribute gepflegt werden können. Zusätzlich wird es gestattet, den Maus-Hover-Stil über CSS festzulegen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

HoverMenu-Komponenten basieren auf div-Elementen und nicht auf Tabellen, weshalb keine Anwendbarkeit dieses Prüfaspekts vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Jedoch können Überlappungen zwischen Menüinhalten und Inhalten der HTML-Seite auftreten, sodass nur noch eine schlechte Lesbarkeit der Inhalte möglich ist. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Untermenüs beim Bewegen des Mauszeigers auf ein Hauptmenü mehr angezeigt werden können.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen Navigations-Elementen, wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.3.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.3.3.4 Symbolleisten

Es wird kein Steuerelement für den Anwendungsfall Symbolleiste im Rahmen des AJAX Control Toolkit ausgeliefert, weshalb alle Teilaspekte mit 0,0 zu bewerten sind.

Bewertung: 0,0

5.3.3.5 Hierarchische Steuerelemente (Tree Views)

Es wird kein spezielles Tree View-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Tree View-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.3.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Fortschrittsanzeige-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.3.3.7 Infotips

5.3.3.7.1 Beschreibung

Es wird keine konkrete Komponente zur Realisierung eines Infotips mit ausgeliefert. Jedoch kann mit vielen Komponenten ein ähnliches Ergebnis erzielt werden. Hierzu soll auf die HoverMenu-Komponente verwiesen werden. Diese stellt die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element einen anderen frei definierbaren Bereich einzublenden, auf welchem beliebige Informationen, z.B. Hilfetexte, positioniert werden können.

5.3.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Da es sich beim HoverMenu um keine vollwertige Infotip-Implementierung handelt, werden auch keine Standard-Layouts für den Inhaltsbereich der Komponente unterstützt. Es wird lediglich ein Container-Element eingeblendet, welches mit beliebigen Elementen füllbar ist, weshalb ein Maximum an Anpassbarkeit vorliegt. Auch Icons und Accesskeys sowie Menüselektionseffekte müssen komplett neu implementiert werden, ohne Hilfe durch diese Komponente. Der Hover-Effekt des InfoTip-Anzeige-auslösenden Elements ist hinsichtlich der Gestaltung anpassbar.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers auf ein beliebiges Element zeigt das entsprechende potentielle Untermenü an. Eine reine Bedienung mit der Tastatur ist browserabhängig nicht oder nur schwer möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Über den ASP.NET-Standarddesigner können InfoTip-Inhalte komfortabel zusammengestellt werden.

Bewertung: 1,0

5.3.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das HoverMenu-Control seine Funktionalität ein. Es werden keine Inhalte im Allgemeinen und Hilfeinhalte im Speziellen mehr angezeigt.

Bewertung: 0,0

5.3.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die Komponenten selbst nur die Anzeige eines Containersteuerelements bei Positionierung des Mauszeigers über einem zu erweiternden Steuerelement implementiert, sodass sämtliche sichtbaren GUI-Elemente in diesem Zusammenhang durch ihre eigenen, originär vorhandenen Style-Attribute gepflegt werden können. Zusätzlich wird es gestattet, den Maus-Hover-Stil über CSS festzulegen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

HoverMenu-Komponenten basieren auf div-Elementen und nicht auf Tabellen, weshalb keine Anwendbarkeit dieses Prüfaspekts vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Jedoch können Überlappungen zwischen Infotip-Inhalten und Inhalten der HTML-Seite auftreten, sodass nur noch eine schlechte Lesbarkeit der Inhalte möglich ist. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Untermenüs beim Bewegen des Mauszeigers auf ein Hauptmenü mehr angezeigt werden können.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.3.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.3.4 Fazit

Das AJAX Control Toolkit bietet viele nützliche, reichhaltige Komponenten, die die Produktpalette von ASP.NET abrunden. Die enthaltenen Komponenten basieren dabei auf den mit ASP.NET 3.5 eingeführten AJAX-Erweiterungen. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.4 ComponentArt Web.UI

5.4.1 Beschreibung

Bei ComponentArt Web.UI handelt es sich nach SYS-CON (2008) um eine Premiumlösung von Benutzeroberflächenkomponenten für ASP.NET. Nach ebd. unterstützt ComponentArt Web.UI mit der aktuellen Version 2008.1 auch die besonders performanten clientseitigen Webserviceaufrufe über clientseitige Proxys (vgl. Esposito, 2007).

Das kommerzielle Produkt ComponentArt Web.UI bietet eine Vielzahl an reichhaltigen Benutzeroberflächenkomponenten wie Registerkartensteuerelemente, Dialoge, Menüs, Symbolleisten und selbst Microsoft Office 2007-Ribbons³⁶.

5.4.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 2008.1 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.4.3 Detailuntersuchung

5.4.3.1 Dialoge

5.4.3.1.1 Beschreibung

ComponentArt Web.UI bietet eine reichhaltige Komponente namens Dialog zur Realisierung von Dialogen (vgl. Abbildung 11). Dieses Steuerelement erlaubt die Einbindung externer Ressourcen oder auch seitenlokaler Inhalte. Es werden verschachtelte Dialoge und Animationen sowie Größenänderungen bei den Dialogen durch Benutzer unterstützt.

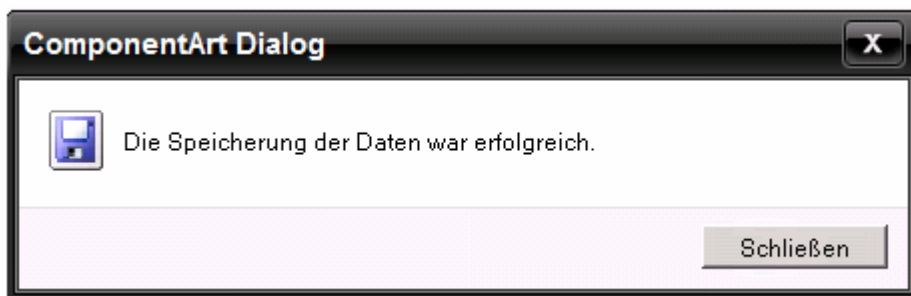


Abbildung 11: Das ComponentArt Web.UI-Dialog-Steuerelement

5.4.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Dialog-Komponente wird standardmäßig nicht über Requestgrenzen hinweg persistiert. Das Ein- und Ausblenden erfolgt per Default über JavaScript.

Bewertung: 0,0

³⁶ Vgl. <http://msdn.microsoft.com/en-us/office/aa905530.aspx>.

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Standardfunktionen wie Dialogvergrößerung durch den Benutzer oder Anzeige von Animationen beim Ein- oder Ausblenden werden ebenso unterstützt. Es liegen jedoch keine über Eigenschaften abrufbare, vorkonfigurierte Layout- und Darstellungsvorlagen vor, um z.B. einen Dialog mit Titelleiste, wie bei Desktopanwendungen, anzuzeigen. Dies ist durch den Entwickler selbst zu erstellen.

Bewertung: 0,5

Verhalten

Es gibt jedoch keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steurelemente zu deklarieren. Auch Standardtastenkürzel für häufige Dialogaktionen, wie das Schließen eines Dialogs, werden nicht ohne Entwicklungsaufwand unterstützt.

Bewertung: 0,0

Entwicklungsunterstützung

Eine Unterstützung liegt bei Benutzung von Visual Studio 2008 vor. Es wird jedoch keine graphische Befüllung des Dialoginhaltes unterstützt, sodass dies in der Codeansicht erfolgen muss.

Bewertung: 0,0

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt eines Dialoges kann von einer externen Ressource bezogen werden. Die direkte Referenzierung von User Controls ist jedoch nur über Umwege möglich.

Bewertung: 0,5

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge unterstützt.

Bewertung: 0,0

5.4.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktion des Ein- und Ausblendens ein.

Bewertung: 0,0

5.4.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header-, Footer- und Body-Bereich sowie weitere Bereiche) eine Anpassung über CSS-Klassen unterstützt wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die umschließenden Rahmen der Dialoge werden über div-Elemente realisiert. Die innere Gestaltung eines Dialogs obliegt dem Entwickler.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente überhaupt nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (z.B. Anzeigen, Verbergen) existiert jedoch keine explizite Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuerelementen um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog-Steuerelemente selbst machen standardmäßig von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden jedoch auch keine spezifischen Elemente, wie z.B. Titelleisten, vorab mitausgeliefert, was als Defizit bewertet werden kann.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.4.3.2 Registerkarten

5.4.3.2.1 Beschreibung

ComponentArt Web.UI unterstützt die Realisierung von Registerkarten durch das TabStrip-Steuerelement (vgl. Abbildung 12). Dieses unterstützt serverseitigen und clientseitigen Tabwechsel, wobei beide Varianten bei nicht verfügbarem JavaScript nicht eingesetzt werden können, da die Komponente als Folge nicht mehr angezeigt wird. Wichtig zu bemerken ist, dass das TabStrip-Steuerelement lediglich die Registerkartenknöpfe ausgibt. Die Inhalte werden von anderen Komponenten, wie z.B. herkömmlichen ASP.NET-MultiPage-Komponenten, bereitgestellt.

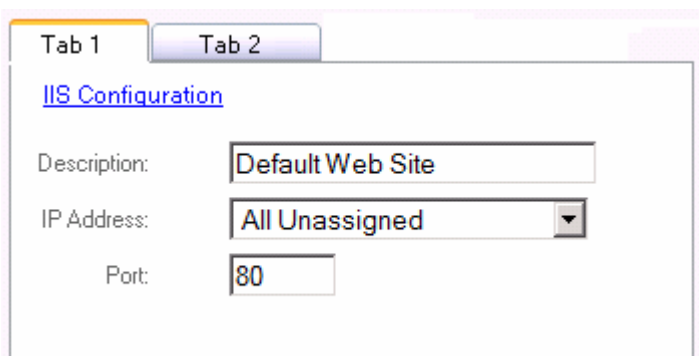


Abbildung 12: Das ComponentArt Web.UI-TabStrip-Steuerelement

5.4.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Tab-Komponente wird über Request-Grenzen hinweg persistiert.

Bewertung: 1,0

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können ebenfalls flexibel, z.B. als Text und Icons, angepasst werden. Zudem wird die Auslagerung der Registerkartenknopfdaten in eine XML-Ressource unterstützt.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist mit Ausnahme von Accesskeys nur äußerst eingeschränkt und nicht browserübergreifend möglich, da die Fokussierung der Tabknöpfe standardmäßig nicht funktioniert.

Bewertung: 0,5

Entwicklungsunterstützung

ComponentArt ermöglicht eine umfangreiche, designergestützte Dialog-Entwicklung an. Weitere Erleichterung wird durch die Verwendung von Smart Tags erreicht.

Bewertung: 1,0

5.4.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Tab-Control seine Funktionalität ein. Es ist kein Tabwechsel mehr möglich, da die Komponente nicht einmal mehr angezeigt wird.

Bewertung: 0,0

5.4.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, CSS-Klassen für die verschiedenen Registerkartenknöpfe vorzugeben.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Tabknöpfe basieren standardmäßig auf Tabellen, obwohl kein legitimer Anwendungsfall für eine Tabelle vorliegt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die

Registerkartenknöpfe können jedoch in den meisten Browsern nicht fokussiert werden. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden jedoch auch keine spezifischen Elemente, wie z.B. Submit-Buttons oder Überschriften als Registerkartenknöpfe verbindlich vorgegeben, was als Defizit zu bewerten ist.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.4.3.3 Menüs

5.4.3.3.1 Beschreibung

Menüs können über das Menu-Steuerelement (vgl. Abbildung 13) realisiert werden. Die Anzeige der Untermenüs eines Menüpunktes erfolgt dabei standardmäßig bei Bewegung des Mauszeigers über den jeweiligen Menüpunkt. Es werden sowohl reine Textmenüpunkte als auch Menüpunkte mit Text und Icons sowie Kontextmenüs unterstützt.

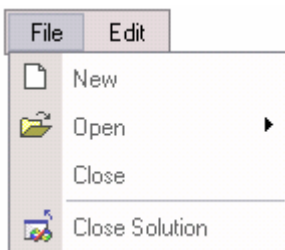


Abbildung 13: Das ComponentArt Web.UI-Menu-Control

5.4.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Es können verschachtelte Menüstrukturen erstellt werden. Neben einfachen Text-Menüpunkten können auch Symbole verwendet werden. Jeder Menüeintrag kann über eine Vielzahl von Einstellungen an die gewünschten Bedürfnisse adaptiert werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine reine Tastatur-Bedienung ist browserabhängig nicht oder nur schwer möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Menükomponenten können extern über eine XML-Ressource hinterlegt werden oder alternativ über einen Visual Studio Designer, den ComponentArt Menu Designer, definiert werden.

Bewertung: 1,0

5.4.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr angezeigt.

Bewertung: 0,0

5.4.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale oder auch menüelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Menu-Komponenten basieren standardmäßig auf Tabellen, was eine Zweckentfremdung der html-Tabellenelemente darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr angezeigt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Entwicklungsunterstützung

Symbolleistenkomponenten können extern über eine XML-Ressource (SiteMap) hinterlegt werden oder alternativ über einen Visual Studio Designer, den ComponentArt ToolBar Designer, komfortabel bearbeitet werden.

Bewertung: 1,0

5.4.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das ToolBar-Control seine Funktionalität komplett ein.

Bewertung: 0,0

5.4.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

ToolBar-Komponenten basieren standardmäßig auf Tabellen, was eine Zweckentfremdung der html-Tabellenelemente darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr angezeigt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist über Shortcuts möglich.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.4.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.4.3.5 Hierarchische Steuerelemente (Tree Views)

5.4.3.5.1 Beschreibung

Hierarchische Strukturen können über das TreeView-Steuerelement (vgl. Abbildung 15) realisiert werden. Dieses dient primär der Anzeige von Navigationsstrukturen, kann aber auch anderweitig beliebig eingesetzt werden. Komfortfunktionen wie Drag & Drop, Checkboxes, Kontextmenüs, Knotentextbearbeitung und Tastaturansteuerung werden ebenfalls unterstützt.

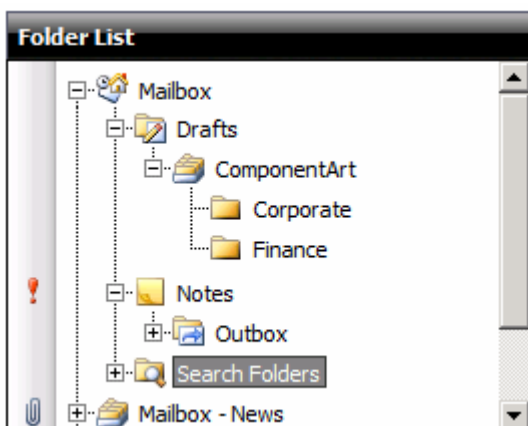


Abbildung 15: Das ComponentArt Web.UI-TreeView-Steuerelement

5.4.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist.

Bewertung: 0,5

Anpassbarkeit

Das TreeView-Steuerelement ist flexibel anpassbar. So können Style-Klassen global festgelegt werden, oder aber auch die Darstellung für jedes einzelne Baumknoten-Element via CSS angepasst werden. Darüber hinaus ist es möglich, einem TreeView-Node (Baumknoten) neben Text auch Symbole hinzuzufügen. Auch eine Anpassung im Hinblick auf die Anzeige von Checkboxes in Baumknoten wird unterstützt.

Bewertung: 1,0

Verhalten

Ein Anklicken eines TreeView-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses, sofern JavaScript aktiviert ist. Unterelemente werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch ebenso nur bei aktiviertem JavaScript. Die anzuzeigenden Elemente können aus einer SiteMap-XML-Datei stammen oder über den ComponentArt TreeView Designer festgelegt werden. Auch eine Selektion der Elemente über die Tastatur wird unterstützt.

Bewertung: 1,0

Entwicklungsunterstützung

Der Aufbau der Baumstrukturen wird durch den sogenannten ComponentArt TreeView Designer auf einfache Art und Weise ermöglicht.

Bewertung: 1,0

5.4.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität ein. Es werden keine Baumknoten angezeigt.

Bewertung: 0,0

5.4.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können Alternativtexte spezifiziert werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Farbenwahl verbleibt in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global diverse graphische Einstellungen vorgenommen werden können. Auch auf Baumknotenebene können DetailEinstellungen hinsichtlich der Visualisierung basierend auf CSS-Klassen vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet eine Zweckentfremdung des table-Elements bei dieser Komponente statt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Darstellung der Knoten mehr erfolgt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird unterstützt.

Bewertung: 1,0

Anforderung 10

Diese Komponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.5.5 Browserinteroperabilität

Das TreeView-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.4.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Fortschrittsanzeige-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.4.3.7 Infotips

5.4.3.7.1 Beschreibung

Es wird keine konkrete Komponente zur Realisierung eines Infotips mit ausgeliefert. Jedoch kann mit vielen Komponenten ein ähnliches Ergebnis erzielt werden. Hierzu soll auf die Dialog-Komponente (vgl. Abbildung 16) näher eingegangen werden. Diese stellt die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element einen anderen frei definierbaren Bereich einzublenden, auf welchem beliebige Informationen, z.B. Hilfetexte positioniert werden können.



Abbildung 16: Das ComponentArt Web.UI-Dialog-Steuerelement als InfoTip

5.4.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Da es sich beim Dialog um keine vollwertige Infotip-Implementierung handelt, werden auch keine Standard-Layouts für den Inhaltsbereich der Komponente unterstützt. Es wird lediglich ein Container-Element eingeblendet, welches mit beliebigen Elementen füllbar ist, weshalb ein Maximum an Anpassbarkeit vorliegt. Auch Icons und Accesskeys sowie Menüselektionseffekte müssen komplett neu implementiert werden, ohne Hilfe durch diese Komponente.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers auf ein beliebiges Element zeigt den entsprechenden Infotip-Inhalt an. Eine reine Bedienung mit der Tastatur ist browserabhängig nicht oder nur schwer möglich. Das Interval, nachdem der InfoTip angezeigt werden soll, ist frei konfigurierbar.

Bewertung: 0,5

Entwicklungsunterstützung

Über den ASP.NET-Standarddesigner können InfoTip-Eigenschaft festgelegt werden. Der Inhalt muss jedoch über die Code-Ansicht eingepflegt werden.

Bewertung: 0,5

5.4.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Dialog-Control seine Funktionalität ein. Es werden keine Inhalte im Allgemeinen und Hilfeinhalte im Speziellen mehr angezeigt.

Bewertung: 0,0

5.4.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die Komponenten selbst nur die Anzeige eines Containersteuerelements bei Positionierung des Mauszeigers über einem zu erweiternden Steuerelement implementiert, sodass sämtliche sichtbaren GUI-Elemente in diesem Zusammenhang durch ihre eigenen, originär vorhandenen Style-Attribute gepflegt werden können.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Dialog-Komponenten basieren auf div-Elementen und nicht auf Tabellen, weshalb keine Anwendbarkeit dieses Prüfaspekts vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was

bedeutet, dass kein Hilfedialog beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Einblendung des Dialogs bzw. Infotips keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog- bzw. InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.4.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.4.4 Fazit

Das ComponentArt Web.UI-Framework bietet viele nützliche, reichhaltige und flexible Komponenten, die die Produktpalette von ASP.NET nicht nur abrunden, sondern wesentlich erweitern. Die enthaltenen Komponenten basieren dabei auf den mit ASP.NET 3.5 eingeführten AJAX-Erweiterungen. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.5 NetAdvantage for ASP. NET

5.5.1 Beschreibung

NetAdvantage for ASP. NET basiert nach Infragistics (2008a) auf dem ASP .NET AJAX-Framework von Microsoft, um moderne Webentwicklungstechnologien bereitzustellen. Nach ebd. stellt die aktuelle Version 2008.1 Entwicklern Werkzeuge zur Verfügung um ein überzeugendes Benutzererlebnis in LOB³⁷-Anwendungen im Anwendungsbereich Web zu erstellen.

Die aktuelle Version integriert sich nahtlos in Visual Studio und das .NET Framework 3.5 und bietet nach ebd. folgende weitere Schlüsselvorteile:

³⁷ LOB = Line of Business; LOB-Anwendungen sind entscheidende, unabdingbare und kritische Software-Applikationen eines Unternehmens.

- Mitauslieferung des Aikido-Frameworks, welches Entwicklern Steuerelemente, die auf einem einzelnen, konsistenten Framework beruhen, zur Verfügung stellt, was die Integration in eigene Applikationen wiederum vereinfacht
- Vereinfachte, XHTML-konforme Ausgabe durch die bereitgestellten Steuerelemente
- CSS-Anpassbarkeit um wesentlich verbesserte und flexiblere Gestaltungsmöglichkeiten für Steuerelemente anzubieten
- Vereinfachtes Deployment mit einer einzigen Laufzeitassembly, welche alle JavaScript-Ressourcen integriert hat

Im aktuellen Release werden nach ebd. zudem neue Komfortkomponenten wie WebDialogWindow unterstützt, womit WinForms-ähnliche Dialoge realisiert werden können. Daneben werden aber auch reguläre Komponenten wie Registerkartensteuerlemene, Menüsteuerelemente und Symbolleisten unterstützt.

Wenngleich die exakte Bezeichnung NetAdvantage for ASP.NET lautet, soll im Rahmen dieser Untersuchung aus Platzgründen stattdessen NetAdvantage ebenso zulässig sein. Das im späteren Verlauf dieser Untersuchung betrachtete NetAdvantage for JSF ist jedoch dadurch abzugrenzen, dass dann stets die vollständige Bezeichnung zu verwenden ist.

5.5.2 Evaluerte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 2008, Volume. 1 aktuell, weshalb sich die Untersuchung primär auf diese Version bezieht.

5.5.3 Detailuntersuchung

5.5.3.1 Dialoge

5.5.3.1.1 Beschreibung

Infragistics NetAdvantage for ASP.NET bietet eine Komponente namens WebDialogWindow zur Realisierung von Dialogen (vgl. Abbildung 17). Dieses Steuerelement unterstützt Funktionen wie Minimierung, Maximierung und Schließen des Dialogs ohne großen Implementierungsaufwand. Auch wird Drag & Drop standardmäßig unterstützt.

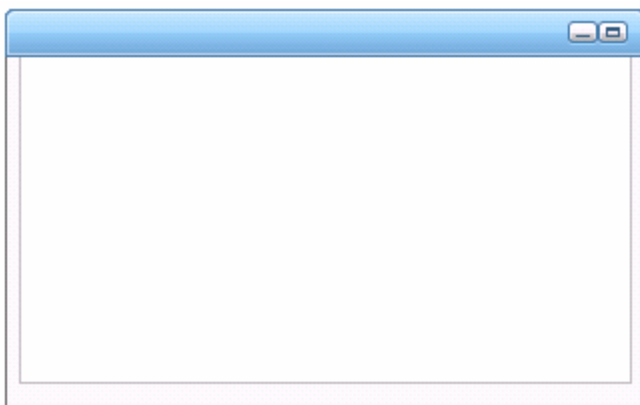


Abbildung 17: Das NetAdvantage-Dialog-Steuerelement

5.5.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Dialog-Komponente wird nur bei aktiviertem JavaScript über Requestgrenzen hinweg persistiert. Bei deaktiviertem JavaScript lassen sich auch die Funktionen wie Ein- und Ausblenden (z.B. über die Schließen-Schaltfläche des Dialogs) nicht mehr nutzen, weshalb auch keine zu persistierenden Zustandsänderungen mehr erzeugt werden können.

Bewertung: 0,5

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Standardfunktionen wie Dialogminimierung, Dialogmaximierung oder auch das Schließen des Dialogs werden unterstützt. Die Grundstruktur und das Layout eines typischen Dialoges werden bereitgestellt. Der Dialog lässt sich über Theming bzw. Stylesheets im Hinblick auf die Titelleiste und auch den Bodybereich flexibel anpassen.

Bewertung: 1,0

Verhalten

Standardfunktionen wie Dialogminimierung, Dialogmaximierung oder auch das Schließen des Dialogs werden unterstützt. Es gibt jedoch keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Auch Standardtastenkürzel für häufige Dialogaktionen, wie das Schließen eines Dialogs, werden nicht ohne Entwicklungsaufwand unterstützt. Es können jedoch clientseitige Ereignisse des Dialogs abgefangen und behandelt werden.

Bewertung: 0,0

Entwicklungsunterstützung

Eine akzeptable Unterstützung liegt bei Benutzung von Visual Studio 2008 vor. Es wird eine graphische Befüllung des Dialoginhaltes unterstützt, sodass dieser nicht in der Codeansicht erfolgen muss. Daneben können primäre Einstellungen über die SmartTag-Unterstützung vorgenommen werden.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt eines Dialoges kann nicht out of the box von einer externen Ressource bezogen werden. Eine Referenzierung über User Controls wäre denkbar, wird jedoch nicht vom Dialog als immanente Eigenschaft selbst unterstützt.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge nativ unterstützt.

Bewertung: 0,0

5.5.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktion des Ein- und Ausblendens ein.

Bewertung: 0,0

5.5.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header-, Footer- und Body-Bereich sowie weitere Bereiche) eine Anpassung über CSS-Klassen unterstützt wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die Dialoge werden standardmäßig über eine Mischung aus div- und table-Elementen realisiert, wobei die Verwendung von table-Elementen eine Zweckentfremdung darstellt. Die innere Gestaltung eines Dialogs obliegt dem Entwickler.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente überhaupt nicht mehr sinnvoll verwendet werden kann, d.h. Ein- und Ausblenden des Dialogs sowie weitere Funktionen wie Minimierung und Maximierung sind dann nicht mehr verfügbar.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (Anzeigen, Verbergen) existiert jedoch keine explizite Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuerelementen um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog-Steuerelemente selbst machen standardmäßig von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation wird primär die Unterteilung in Dialog-Titelbarleiste und Dialog-Inhalt gegeben. Eine weitergehende Unterstützung liegt nicht vor.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von

zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.5.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.5.3.2 Registerkarten

5.5.3.2.1 Beschreibung

NetAdvantage for ASP.NET unterstützt die Realisierung von Registerkarten durch das UltraWebTab-Steuerelement (vgl. Abbildung 18). Dieses unterstützt clientseitigen Tabwechsel, was jedoch nur bei aktiviertem JavaScript eingesetzt werden kann. Sowohl die Tabknöpfe als auch die Tabinhalte (Containersteuerelemente) werden über die Komponente bereitgestellt.

Customer List Customer Info Order Information		
Drag a column header here to group by that column		
CustomerID ▾	CompanyName	ContactTitle
ALFKI	Alfreds Futterkiste	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Owner
ANTON	Antonio Moreno Taquería	Owner
AROUT	Around the Horn	Sales Representative
BERGS	Berglunds snabbköp	Order Administrator
BLAUS	Blauer See Delikatessen	Sales Representative

Abbildung 18: Das NetAdvantage-UltraWebTab-Steuerelement

5.5.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Tab-Komponente wird über Request-Grenzen hinweg persistiert. Eine zu persistierende Zustandsänderung kann jedoch nur bei aktiviertem JavaScript ausgelöst werden, sodass eine Persistierung letztendlich auch nur bei aktiviertem JavaScript vorgenommen wird.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können ebenfalls hinsichtlich der Darstellung flexibel angepasst werden. Eine Anpassung ist auf Basis direkter Stilvorgeben oder auf Basis von CSS-Klassen möglich.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist mit Ausnahme von Accesskeys nur äußerst eingeschränkt und nicht browserübergreifend möglich, da die Fokussierung der Tabknöpfe standardmäßig nicht funktioniert.

Bewertung: 0,5

Entwicklungsunterstützung

Infragistics NetAdvantage for ASP.NET ermöglicht eine umfangreiche, designergestützte Dialog-Entwicklung an. Weitere Erleichterung wird durch die Verwendung von Smart Tags erreicht, worüber unter Anderem einige Darstellungseigenschaften der Komponente auf einfache Art und Weise angepasst werden können. Die Registerkarten können über den Designer durch Umschalten der Tabs und Drag & Drop von Steuerelementen zur Designzeit befüllt werden.

Bewertung: 1,0

5.5.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Tab-Control seine Funktionalität ein. Auch Funktionalität wie serverseitiges AutoPostBack ist dann nicht mehr verfügbar, da dieses auf clientseitige Aufrufe zurückzuführen ist.

Bewertung: 0,0

5.5.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, CSS-Klassen für Registerkartenknöpfe vorzugeben.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Tabknöpfe basieren standardmäßig auf Tabellen, obwohl kein legitimer Anwendungsfall für eine Tabelle vorliegt. Zudem werden iframe-Elemente verwendet.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr sinnvoll verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können jedoch in den meisten Browsern nicht fokussiert werden. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst benutzen iframe-Elemente. Diese werden jedoch nicht dazu benutzt um fremde Inhalte für den Dialog zu laden sondern als Workaround-Lösung eingesetzt, also nicht zur Erfüllung der originären Aufgabe von frame- bzw. iframe-Elementen. Zur verbesserten Orientierung und Navigation werden auch keine spezifischen Elemente, wie z.B. Submit-Buttons oder Überschriften als Registerkartenknöpfe verbindlich vorgegeben, was als Defizit zu bewerten ist.

Bewertung: 0,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von

zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.5.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.5.3.3 Menüs

5.5.3.3.1 Beschreibung

Menüs können über das UltraWebMenu-Steuerelement (vgl. Abbildung 19) realisiert werden. Die Darstellung der Menüs kann flexibel gestaltet werden. Über den sogenannten UltraWebMenu Designer erfolgt die komfortable Vorgabe aller Menüelemente. Eine Bindung an eine Sitemap oder eine XML-Ressource wird unterstützt.

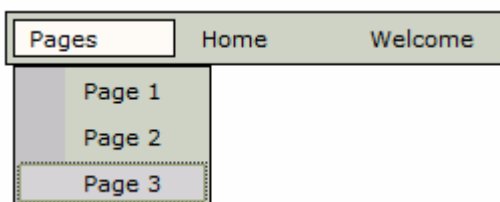


Abbildung 19: Das NetAdvantage-UltraWebMenu-Control

5.5.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Es können verschachtelte Menüstrukturen erstellt werden. Neben einfachen Text-Menüpunkten können auch Symbole und Hover-Symbole verwendet werden. Jeder Menüeintrag kann über eine Vielzahl von Einstellungen an die gewünschten Bedürfnisse adaptiert werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine reine Bedienung mit der Tastatur ist nur bei aktiviertem JavaScript möglich, wenn man von der reinen Accesskey-Implementierung absieht.

Bewertung: 0,5

Entwicklungsunterstützung

Menükomponenten können extern über eine XML-Ressource hinterlegt werden oder alternativ über einen Visual Studio Designer, den WebMenu Designer, komfortabel definiert werden.

Bewertung: 1,0

5.5.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität der Anzeige untergeordneter Menüelemente ein.

Bewertung: 0,0

5.5.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte verwendet werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale oder auch menüelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Menu-Komponenten basieren standardmäßig auf Tabellen, was eine Zweckentfremdung der html-Tabellenelemente darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung trotz geändertem Layout weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr angezeigt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist im Fall von JavaScript-Verfügbarkeit vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.5.3.3.5 Browserinteroperabilität

Die Browserinteroperabilität der Komponente stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Untermenüelemente werden bei Bewegung der Maus über die Top-Level-Menüelemente nicht angezeigt.

Bewertung: 0,0

Safari 3.1+

Untermenüelemente werden bei Bewegung der Maus über die Top-Level-Menüelemente nicht angezeigt.

Bewertung: 0,0

5.5.3.4 Symbolleisten

5.5.3.4.1 Beschreibung

Symbolleisten können über das Steuerelement UltraWebToolBar (vgl. Abbildung 20) realisiert werden. Dabei werden als UltraWebToolBar-Komponenten unter Anderem einfache Buttons, Textfelder, Labels, Separatoren und benutzerdefinierte Steuerelemente unterstützt.

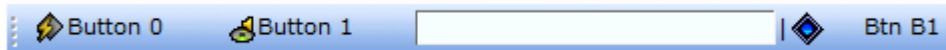


Abbildung 20: Das NetAdvantage-UltraWebToolBar-Control

5.5.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Das UltraWebToolBar-Control von Infragistics unterstützt eine Vielzahl verschiedener Symbolleistenschaltflächentypen. Dazu zählen, Buttons, Textfelder, Labels, Separatoren, wobei über die Eigenschaften dieser Elemente weitere Differenzierungen möglich sind. So kann die Button-Komponente hinsichtlich des Verhaltens z.B. auch als CheckBox verwendet werden. Auch können benutzerdefinierte Elemente auf der Symbolleiste platziert werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers zeigt das entsprechende Icon im festgelegten Hover-Zustand an. Ein Klick darauf führt die hinterlegte Aktion aus bzw. setzt den Selektionszustand der jeweiligen Komponente. Accesskeys werden unterstützt, echte Tastaturkombinationen hingegen nicht.

Bewertung: 0,5

Entwicklungsunterstützung

Symbolleistenkomponenten können komfortabel über einen Visual Studio-Dialog, den Edit ToolBar-Dialog, bearbeitet werden.

Bewertung: 1,0

5.5.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das ToolBar-Control seine Funktionalität komplett ein. So kann der Selektionszustand von Toolbar-Buttons nicht mehr geändert werden und auch die Auslösung von serverseitigen Ereignissen bei Anklicken eines Toolbar-Buttons ist nicht mehr möglich.

Bewertung: 0,0

5.5.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte

spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

ToolBar-Komponenten basieren standardmäßig auf Tabellen, was eine Zweckentfremdung der html-Tabellenelemente darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr auf Benutzeraktionen reagiert.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist nur über Accesskeys, nicht aber über Shortcuts, möglich.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.5.3.4.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor, wenngleich die Standard-Darstellung von der korrekten (vgl. Internet Explorer 6.0 / Mozilla Firefox 2.0) abweicht. Dies lässt sich jedoch über CSS kompensieren.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor, wenngleich die Standard-Darstellung von der korrekten (vgl. Internet Explorer 6.0 / Mozilla Firefox 2.0) abweicht. Dies lässt sich jedoch über CSS kompensieren.

Bewertung: 1,0

5.5.3.5 Hierarchische Steuerelemente (Tree Views)

5.5.3.5.1 Beschreibung

Hierarchische Strukturen können über das UltraWebTree-Steuerelement (vgl. Abbildung 21) realisiert werden. Dieses dient der Anzeige von Navigationsstrukturen, kann aber auch anderweitig beliebig eingesetzt werden. Komfortfunktionen wie Drag & Drop, Checkboxes, Knotentextbearbeitung und Tastaturansteuerung werden ebenfalls unterstützt, wenngleich nicht durchgängig browserinteroperabel.

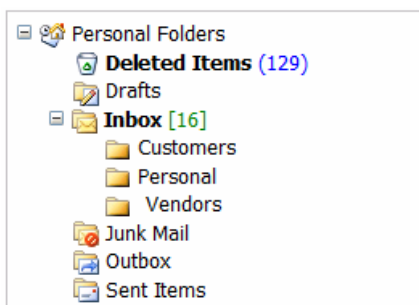


Abbildung 21: Das NetAdvantage-UltraWebTree-Steuerelement

5.5.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist.

Bewertung: 0,5

Anpassbarkeit

Das UltraWebTree-Steuerelement ist flexibel anpassbar. So können Style-Klassen global festgelegt werden, oder aber auch die Darstellung für jedes einzelne Baumknoten-Element via CSS angepasst werden. Darüber hinaus ist es möglich, einem Baumknoten neben Text auch Symbole hinzuzufügen. Auch eine Anpassung im Hinblick auf die Anzeige von Checkboxes in Baumknoten wird unterstützt.

Bewertung: 1,0

Verhalten

Ein Anklicken eines Baumknotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses, sofern JavaScript aktiviert ist. Unterelemente werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch ebenso nur bei aktiviertem JavaScript. Die anzuzeigenden Elemente können komfortabel über den NetAdvantage WebTree Designer festgelegt werden. Auch eine Selektion der Elemente über die Tastatur wird unterstützt.

Bewertung: 1,0

Entwicklungsunterstützung

Der Aufbau der Baumstrukturen wird durch den sogenannten WebTree Designer auf einfache Art und Weise ermöglicht.

Bewertung: 1,0

5.5.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität ein. Die Kollabierung bzw. Expansion von Baumknoten sowie die Ausführung von Aktionen ist dann nicht mehr verfügbar.

Bewertung: 0,0

5.5.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können Alternativtexte spezifiziert werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung strikt vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global diverse graphische Einstellungen vorgenommen werden können. Auch auf Baumknotenebene können Detailsinstellungen hinsichtlich der Visualisierung basierend auf CSS-Klassen vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die Ausgabe der Komponente erfolgt über div- und span-Komponenten.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung der Knoten mit der Tastatur wird nur bei aktiviertem JavaScript browserübergreifend unterstützt.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die UltraWebTree-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht vorgesehen. Beispielsweise kommen statt Listenstrukturierungselementen div- und span-Elemente zum Einsatz.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.5.3.5.5 Browserinteroperabilität

Das TreeView-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine primären funktionalen Einschränkungen vor, wenngleich erweiterte optionale Funktionen wie Drag & Drop nicht oder eingeschränkt funktionieren.

Bewertung: 1,0

Opera 9+

Es liegen keine primären funktionalen Einschränkungen vor, wenngleich erweiterte optionale Funktionen wie Drag & Drop nicht oder eingeschränkt funktionieren.

Bewertung: 1,0

Safari 3.1+

Es liegen keine primären funktionalen Einschränkungen vor, wenngleich erweiterte optionale Funktionen wie Drag & Drop nicht oder eingeschränkt funktionieren.

Bewertung: 1,0

5.5.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Fortschrittsanzeige-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.5.3.7 Infotips

5.5.3.7.1 Beschreibung

Es wird keine konkrete Komponente zur Realisierung eines Infotips mit ausgeliefert.

Gesamtbewertung: 0,0

5.5.4 Fazit

Das Infragistics NetAdvantage for ASP.NET-Framework bietet viele reichhaltige und flexible Komponenten, die die Produktpalette von ASP.NET nicht nur abrunden, sondern wesentlich erweitern. Wenngleich im großen und ganzen gut und unkritisch, besitzen die Aspekte Browserinteroperabilität ebenso wie Barrierefreiheit Verbesserungspotentiale. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.6 Telerik RadControls for ASP.NET AJAX

5.6.1 Beschreibung

Bei Telerik RadControls for ASP.NET AJAX handelt es sich nach Telerik (2008) um eine umfassende Werkzeugsammlung mit über 20 marktführenden Steuerelementen, welche Reichhaltigkeit und Interaktivität für eigene Webapplikationen mit sich bringen. Nach ebd. sollen sich auch Entwicklungszeiten durch ein einheitliches Programmiermodell und umfangreiche Design Time-Unterstützung deutlich senken lassen.

Das kommerzielle Telerik RadControls basiert in der aktuellen Version auf ASP.NET AJAX und nutzt dabei die entsprechenden Dienste dieser Basisplattform. Auch die Performance soll nach ebd. trotz des mächtigen Funktionsumfangs als sehr positiv zu bewerten sein, was auch an Mechanismen wie dem Nachladen von Skriptressourcen on demand liegt.

5.6.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 2008.1 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.6.3 Detailuntersuchung

5.6.3.1 Dialoge

5.6.3.1.1 Beschreibung

RadControls for ASP.NET AJAX bietet eine reichhaltige Komponente namens RadWindow zur Realisierung von Dialogen (vgl. Abbildung 22). Dieses Steuerelement erlaubt die Anzeige von Dialogen, welche Komfortfunktionen wie Drag & Drop-Verschiebung, Größenänderung, Minimierung, Maximierung und Verankerung sowie Animationen unterstützt. Über den RadWindowManager können Einstellungen bzw. Aktionen auf alle RadWindow-Objekte der aktuellen Webform angewendet werden.

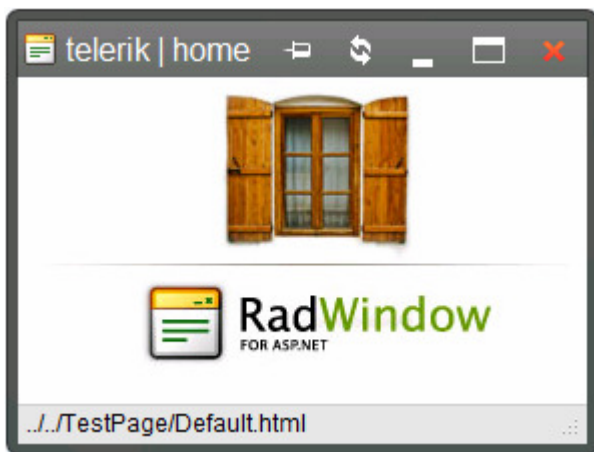


Abbildung 22: Das RadWindow-Steuerelement

5.6.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Dialog-Komponente wird standardmäßig nicht über Requestgrenzen hinweg persistiert. Das Ein- und Ausblenden erfolgt per Default über JavaScript. Es wird eine serverseitige Datenübergabe unterstützt, die jedoch auch nur bei aktiviertem JavaScript zuverlässig funktioniert.

Bewertung: 0,0

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Das Steuerelement erlaubt die einfache Einstellung über Steuerelementeigenschaften, welche Komfortfunktionen, wie Drag & Drop-Verschiebung, Größenänderung, Minimierung, Maximierung und Verankerung sowie Animationen, verwendet werden sollen. Die Darstellung lässt sich über Skins anpassen.

Bewertung: 1,0

Verhalten

Es gibt keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Lediglich eine Komponente kann über die Eigenschaft `OpenerElementID` deklarativ als Öffnen-Steuerelement festgelegt werden. Auch Standardtastenkürzel für häufige Dialogaktionen, wie das Schließen eines Dialogs, werden nicht ohne Entwicklungsaufwand unterstützt. Grundsätzlich wird aber vom Funktionsumfang her eine Vielzahl an verschiedenen Funktionen (siehe obigen Abschnitt Anpassbarkeit) unterstützt.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung innerhalb vom Visual Studio 2008 bereitgestellt. Dies umfasst SmartTag-Unterstützung, Befüllung des Dialoginhalts via Drag & Drop sowie die Möglichkeit sämtliche Eigenschaften über das Visual Studip-Eigenschaftsfenster komfortabel vorgeben zu können.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt eines Dialoges kann nicht automatisch von einer externen Ressource bezogen werden. Hierzu wäre eine benutzerdefinierte Implementierung zusätzlich erforderlich.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert eine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen. Dies ist derzeit jedoch nur über JavaScript implementiert.

Bewertung: 0,5

Standard-Dialoge für MessageBoxen

Es werden als Standarddialoge eine Alert-, Confirm- und Prompt-Box unterstützt. Damit lassen sich einfache Benachrichtigungen an den Benutzer, Rückfragen oder selbst einfache Aufforderungen an den Benutzer mit Dateneingabemöglichkeit auf einfache und schnelle Art realisieren. Die Anzeige der Dialoge erfolgt jedoch derzeit nur via JavaScript.

Bewertung: 0,5

5.6.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie Ein- und Ausblenden, ein.

Bewertung: 0,0

5.6.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header-, Footer- und Body-Bereich sowie weitere Bereiche) eine Anpassung über CSS-Klassen bzw. Skins unterstützt wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die umschließenden Rahmen der Dialoge werden über Tabellenelemente realisiert, was als negativ zu bewerten ist. Die innere Gestaltung eines Dialogs obliegt dem Entwickler. Durch Skins lassen sich jedoch weitgehend Anpassungen vornehmen.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann überhaupt nicht mehr dargestellt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (z.B. Anzeigen, Verbergen) existiert jedoch keine explizite Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuerelementen um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog-Steuerelemente selbst verwenden iframe-Elemente für die Darstellung der Dialoginhalte. Frametitel sind nicht zu erkennen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegt in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.6.3.2 Registerkarten

5.6.3.2.1 Beschreibung

Telerik unterstützt die Realisierung von Registerkarten durch das RadTabStrip-Steuerelement (vgl. Abbildung 23). Wichtig ist hierbei, dass das RadTabStrip-Steuerelement lediglich die Registerkartenknöpfe rendert. Als Containersteuerelement für die Registerkarteninhalte fungiert ein RadTabMultiPage-Steuerelement. Somit bewirkt die Auswahl eines RadTabStrip-Tabs die Anzeige der zugehörigen RadTabMultiPage. Das RadTabStrip-Steuerelement unterstützt unter anderem auch hierarchische Registerkarten und bietet umfangreiche Databinding-Möglichkeiten.

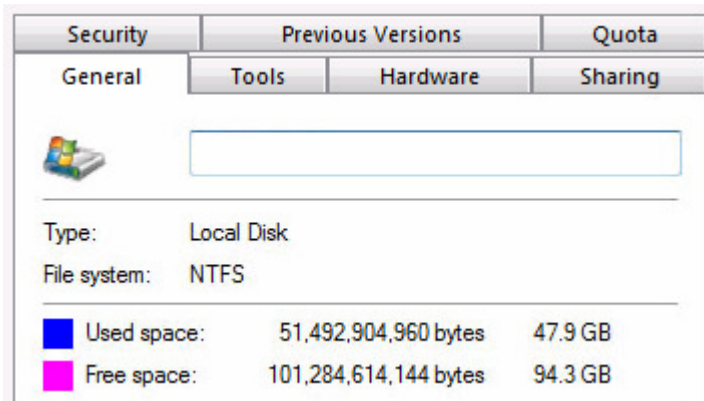


Abbildung 23: Das RadTabStrip-Steuerelement und das RadMultiPage-Steuerelement

5.6.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Tab-Komponente wird über Request-Grenzen hinweg persistiert, falls JavaScript aktiviert ist. Bei deaktiviertem JavaScript ist auch kein Registerkartenwechsel ohne Weiteres möglich, sodass die Betrachtung der Request-Lebenszyklus-Persistenz in diesem Fall zu verneinen ist.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können flexibel, auch über eigens dafür existierende Templates, angepasst werden. Auch DataBinding kann als dynamische Grundlage für Anpassungen der Registerkarten dienen.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist mit Ausnahme von Accesskeys nur äußerst eingeschränkt und nicht browserübergreifend möglich, da die Fokussierung der Tabknöpfe standardmäßig nur bei aktiviertem bzw. verfügbarem JavaScript funktioniert.

Bewertung: 0,5

Entwicklungsunterstützung

Es besteht eine umfangreiche, designergestützte Registerkartenentwicklung. Besonders hervorzuheben ist in diesem Zusammenhang der RadTabStrip Item Builder, über welchen auch hierarchische Tabstrukturen komfortabel aufgebaut werden können. Auch können Anpassungen an der Darstellung einzelner Registerkartenknöpfe hierüber erfolgen. Weitere Erleichterung wird durch die Unterstützung von Smart Tags erreicht.

Bewertung: 1,0

5.6.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Tab-Control seine Funktionalität ein. Es ist standardmäßig kein Tabwechsel mehr möglich.

Bewertung: 0,0

5.6.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS global oder auch registerkartenknopflokale Anpassungen an der Darstellung vorzunehmen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig auf Div- und Auflistungselementen, was im Sinne der Barrierefreiheit als positiv zu bewerten ist.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können jedoch in den meisten Browsern nicht fokussiert werden, falls JavaScript deaktiviert ist. Tastaturkürzel (Accesskeys) werden unterstützt.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden HTML-Listenelemente verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden HTML-Listenelemente verwendet.

Bewertung: 1,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.6.3.3 Menüs

5.6.3.3.1 Beschreibung

Menüs können über das RadMenu-Steuerelement (vgl. Abbildung 24) realisiert werden. Die Anzeige der Untermenüs eines Menüpunktes erfolgt dabei standardmäßig bei Bewegung des Mauszeigers über den jeweiligen Menüpunkt. Es werden sowohl reine Textmenüpunkte als auch Menüpunkte mit Text und Icons sowie Kontextmenüs (über die Komponente RadContextMenu) unterstützt.

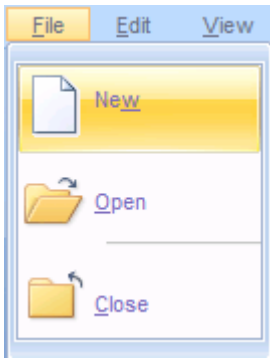


Abbildung 24: Das RadMenu-Control

5.6.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Es können verschachtelte Menüstrukturen erstellt werden. Neben einfachen Text-Menüpunkten können auch Symbole verwendet werden. Jeder Menüeintrag kann über eine Vielzahl von Einstellungen an die gewünschten Bedürfnisse adaptiert werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine komfortable Bedienung mit der Tastatur ist nur bei aktivierter JavaScript-Unterstützung möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Menükomponenten können über einen Visual Studio Designer, den RadMenu Item Builder, komfortabel definiert werden. Daneben verfügt die Komponente über SmartTag-Unterstützung.

Bewertung: 1,0

5.6.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr angezeigt.

Bewertung: 0,0

5.6.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale (über die Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften des jeweiligen Menüelements) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Menu-Komponenten basieren standardmäßig auf Html-Listenelementen, was im Sinne der Barrierefreiheit als positiv zu bewerten ist.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr auswählbar sind.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur ist nur bei aktiviertem JavaScript verfügbar.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung der Navigation werden Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Durch die Verwendung von Html-Listenelementen wird jedoch die grundsätzliche Dokumentstruktur explizit ersichtlich gemacht.

Bewertung: 1,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.6.3.4 Symbolleisten

5.6.3.4.1 Beschreibung

Symbolleisten können über das Steuerelement RadToolBar (vgl. Abbildung 25) realisiert werden. Dabei werden als RadToolBar-Elemente unter Anderem Schaltflächen (mit Text und / oder Symbol), Dropdownmenüs sowie Radio- und Checkboxbuttons (im entsprechenden symbolleistenkonformen Stil) unterstützt.



Abbildung 25: Das RadToolBar-Control

5.6.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Das RadToolBar-Control unterstützt eine Vielzahl verschiedener Symbolleistenschaltflächentypen. Dazu zählen unter Anderem Schaltflächen (mit Text und/oder Symbol sowie Selektionszustandsunterstützung) und Dropdownmenüs. Die Elemente können tiefgreifend angepasst werden, wozu auch die Festlegung der Bilder, Texte, Menüelementgrößen, Accesskeys und Hover-Effekte gehört.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers zeigt das entsprechende Icon im festgelegten Hover-Zustand an. Ein Klick darauf führt bei aktiviertem JavaScript die hinterlegte Aktion aus bzw. setzt den Selektionszustand der jeweiligen Komponente. Tastaturkombinationen werden über Accesskeys unterstützt. Bei aktiviertem JavaScript kann auch mit Cursortasten bzw. der Tabtaste über die Symbolleistenicons navigiert werden.

Bewertung: 0,5

Entwicklungsunterstützung

Symbolleistenkomponenten können über einen Visual Studio Designer, den RadToolBar Item Builder komfortabel bearbeitet werden. Zudem besteht SmartTag-Unterstützung.

Bewertung: 1,0

5.6.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das ToolBar-Control seine Funktionalität komplett ein.

Bewertung: 0,0

5.6.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

ToolBar-Komponenten basieren auf Html-Listenelementen, was im Sinne der Barrierefreiheit als positiv zu bewerten ist.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr angezeigt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist über Accesskeys und bei aktiviertem JavaScript zusätzlich über die Tabulatortaste bzw. die Cursortasten möglich.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur Verbesserung der Strukturierung und Navigation werden Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden jedoch Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.4.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.6.3.5 Hierarchische Steuerelemente (Tree Views)

5.6.3.5.1 Beschreibung

Hierarchische Strukturen können über das RadTreeView-Steuerelement (vgl. Abbildung 26) realisiert werden. Über dieses können Navigationsstrukturen dargestellt werden, oder aber einfach hierarchische Daten visualisiert werden. Komfortfunktionen wie Drag & Drop, Checkboxes, Knotentextbearbeitung, Tastaturansteuerung und Animationen werden ebenfalls unterstützt.

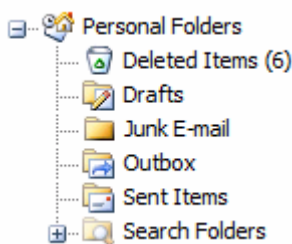


Abbildung 26: Das RadTreeView-Steuerelement

5.6.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist.

Bewertung: 0,5

Anpassbarkeit

Das RadTreeView-Steuerelement ist flexibel anpassbar. So können Style-Klassen global festgelegt werden, oder aber auch die Darstellung für jedes einzelne Baumknoten-Element via

CSS angepasst werden. Darüber hinaus ist es möglich, einem TreeView-Node (Baumknoten) neben Text auch Symbole hinzuzufügen. Auch eine Anpassung im Hinblick auf die Anzeige von Checkboxes in Baumknoten wird unterstützt.

Bewertung: 1,0

Verhalten

Ein Anklicken eines RadTreeView-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses, sofern JavaScript aktiviert ist. Unterelemente werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch ebenso nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Die im RadTreeView anzuzeigenden Elemente können über den Rad TreeView Item Builder komfortabel festgelegt werden. Auch SmartTag-Unterstützung besteht.

Bewertung: 1,0

5.6.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität ein. Ein Expandieren bzw. Kollabieren von Baumknoten ist dann nicht mehr verfügbar.

Bewertung: 0,0

5.6.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können keine spezifischen Alternativtexte vorgegeben werden.

Bewertung: 0,5

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global diverse graphische Einstellungen vorgenommen werden können. Auch auf Baumknotenebene können Detaileinstellungen hinsichtlich der Visualisierung basierend auf CSS-Klassen vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet keine Zweckentfremdung des table-Elements bei dieser Komponente statt, da hierarchische Navigationsstrukturen über Html-Listenelemente abgebildet werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Knotenkollabierung bzw. Knotenexpansion und keine Ausführung von Befehlen mehr erfolgt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird unterstützt, falls JavaScript aktiviert ist.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung der Navigation werden jedoch Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.5.5 Browserinteroperabilität

Das TreeView-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor, wenngleich die Standarddarstellung sich von der vorgesehenen unterscheidet.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor, wenngleich die Standarddarstellung sich von der vorgesehenen unterscheidet.

Bewertung: 1,0

5.6.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Fortschrittsanzeige-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.6.3.7 Infotips

5.6.3.7.1 Beschreibung

Zur Realisierung eines Infotips soll auf die RadToolTip-Komponente (vgl. Abbildung 27) näher eingegangen werden. Diese stellt die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element einen anderen frei definierbaren Bereich einzublenden, auf welchem beliebige Informationen, z.B. Hilfetexte positioniert werden können. Auch Animationen und Load On Demand werden unterstützt.

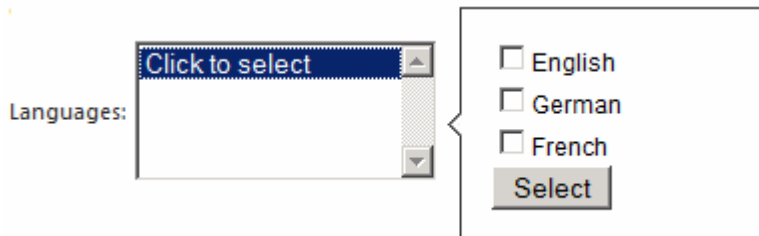


Abbildung 27: Die RadToolTip-Komponente

5.6.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Es werden keine Standard-Layouts für den Inhaltsbereich der Komponente unterstützt. Es wird lediglich ein Container-Element eingeblendet, welches mit beliebigen Elementen füllbar ist, weshalb ein Maximum an Anpassbarkeit vorliegt. Auch Icons und Accesskeys sowie Menüselektionseffekte für enthaltene Elemente müssen komplett neu implementiert werden, ohne besondere Unterstützung durch diese Komponente.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers auf ein entsprechendes Element zeigt den entsprechenden Infotip-Inhalt an. Dieser lässt sich jedoch auch über Codeaufrufe anzeigen. Das Intervall, nachdem der InfoTip und wie lange er angezeigt werden soll, ist frei konfigurierbar.

Bewertung: 1,0

Entwicklungsunterstützung

Über den ASP.NET-Standarddesigner können RadToolTip-Steuerelement-Eigenschaften festgelegt werden. Der Inhalt kann über die Text-Eigenschaft festgelegt werden. Komplexe Inhalte können als Kindelemente des Steuerelements aufgenommen werden.

Bewertung: 1,0

5.6.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität ein. Es werden keine Inhalte im Allgemeinen und Hilfeinhalte im Speziellen mehr angezeigt.

Bewertung: 0,0

5.6.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die Komponenten selbst nur die Anzeige eines Containersteuerelements bei Positionierung des Mauszeigers über einem zu erweiternden Steuerelement implementiert, sodass sämtliche sichtbaren GUI-Elemente in diesem Zusammenhang durch ihre eigenen, originär vorhandenen Style-Attribute gepflegt werden können.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

RadToolTip-Komponenten basieren zur Callout-Darstellung standardmäßig auf Tabellenelementen. Dies lässt sich jedoch umgehen und eine barrierefreie Darstellung stattdessen realisieren.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass kein Infotip beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Einblendung des Dialogs bzw. Infotips keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur unabhängig von der Verfügbarkeit von JavaScript ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog- bzw. InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.6.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.6.4 Fazit

Das Telerik RadControls for ASP.NET AJAX-Framework bietet viele nützliche, reichhaltige und flexible Komponenten, die die Produktpalette von ASP.NET nicht nur abrunden, sondern wesentlich erweitern. Die enthaltenen Komponenten basieren dabei auf den mit ASP.NET 3.5 eingeführten AJAX-Erweiterungen. Die Telerik-Produktsuite stellt die bisher am weitesten entwickelte und auf den Aspekt Barrierefreiheit durchdachtste Lösung dar. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.7 Developer Express ASPxperience Suite

5.7.1 Beschreibung

Bei Developer Express ASPxperience Suite handelt es sich nach Developer Inc. (2008a) um eine Next Generation-Werkzeugsammlung um eigene Webanwendungen mit fortschrittlichen Funktionalitäten unter Anderem in den Bereichen Navigation, Datenflusskontrolle und Datenlayoutverwaltung anzureichern.

Ferner werden nach Developer Inc. (2008b) über 20 Komponenten und alle relevanten Browser unterstützt. Zu den unterstützten Browsern zählen dabei nach ebd. im Einzelnen Internet Explorer 5.5 +, Netscape 7.2 +, Mozilla, Mozilla Firefox, Opera 8+ und Safari. Als Komponenten werden unter Anderem Registerkartensteuerelemente, Dialoge, Menükomponenten und Navigationssteuerelemente zur Verfügung gestellt.

5.7.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 2008.1.3 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.7.3 Detailuntersuchung

5.7.3.1 Dialoge

5.7.3.1.1 Beschreibung

ASPxperience bietet eine reichhaltige Komponente namens ASPxPopupControl zur Realisierung von Dialogen (vgl. Abbildung 28). Wenngleich das Element allgemein zur dynamischen Anzeige von Inhalten wie auch erweiterten Tooltips verwendet werden kann, erlaubt das Steuerelement die Anzeige von Dialogen, welche Komfortfunktionen wie Drag & Drop-Verschiebung, Dialoggrößenänderung, Anzeige eines Dialogschattens sowie Laden anderer Webseiten in den Dialog unterstützen. Die Trigger-Komponente zur Anzeige des Dialogs sowie das Ereignis (z.B. Mausbewegung, Mausklick) sind über die Steuerelementeigenschaften spezifizierbar. Auch wird DataBinding unterstützt.

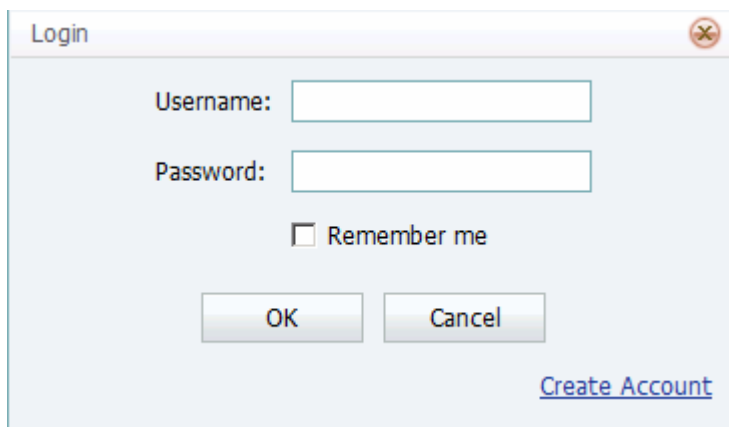


Abbildung 28: Das ASPxPopupControl-Steuerelement

5.7.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Dialog-Komponente wird standardmäßig über Requestgrenzen hinweg persistiert. Dies erfolgt jedoch nur bei aktiviertem JavaScript korrekt. Es existiert kein dialogoriginärer Mechanismus zum Datenaustausch zwischen Dialog und der dialogöffnenden Umgebung.

Bewertung: 0,5

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Das Steuerelement erlaubt die einfache Einstellung über Steuerelementeigenschaften, welche Komfortfunktionen, wie Drag & Drop-Verschiebung, Größenänderung sowie Animationen, verwendet werden sollen. Die Darstellung lässt sich komplett über CSS anpassen. Dabei können den verschiedenen Dialogbestandteilen separat CSS-Klassen bzw. direkt CSS-Attributwerte zugewiesen werden.

Bewertung: 1,0

Verhalten

Es gibt keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Lediglich eine Komponente kann über die Eigenschaft `PopupElementID` deklarativ als Öffnen-Steuerelement festgelegt werden. Jedoch kann das beim Öffnen-Steuerelement aufzutretende Ereignis für eine Dialoganzeige komfortabel über die Steuerelementeigenschaften festgelegt werden. Standardtastenkürzel für häufige Dialogaktionen, wie das Schließen eines Dialogs, werden nicht ohne Entwicklungsaufwand unterstützt. Grundsätzlich wird aber vom Funktionsumfang her eine Vielzahl an verschiedenen Funktionen (siehe obigen Abschnitt Anpassbarkeit) unterstützt.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung innerhalb vom Visual Studio 2008 bereitgestellt. Dies umfasst SmartTag-Unterstützung, Befüllung des Dialoginhalts via Drag & Drop sowie die Möglichkeit sämtliche Eigenschaften über das Visual Studio-Eigenschaftfenster komfortabel vorgeben zu können.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Dialoginhalt eines Dialoges kann von einer externen Ressource geladen werden. Es liegt jedoch keine dialogoriginäre Implementierung zum kontrollierten Datenaustausch zwischen Dialoginhalt und dialogöffnender Umgebung vor.

Bewertung: 0,5

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge ohne eigenen Entwicklungs- bzw. Anpassungsaufwand unterstützt.

Bewertung: 0,0

5.7.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein. Dies trifft auch für den Prozess der Requestpersistierung zu.

Bewertung: 0,0

5.7.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header-, Footer- und Content-Bereich sowie weitere Bereiche) eine Anpassung über CSS-Klassen bzw. Skins unterstützt wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Der Dialog wird per Default über Tabellenelemente realisiert. Die innere Gestaltung eines Dialogs obliegt dem Entwickler.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann überhaupt nicht mehr dargestellt wird.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (Anzeigen, Verbergen) existiert jedoch keine explizite, dialogoriginäre Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuerelementen um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog-Steuerelemente selbst verwenden optional iframe-Elemente zur Einbettung fremder Ressourcen. Frametitel sind nicht zu erkennen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.7.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.7.3.2 Registerkarten

5.7.3.2.1 Beschreibung

ASPxperience unterstützt die Realisierung von Registerkarten durch das ASPxTabControl-Steuererelement (vgl. Abbildung 29). Das ASPxTabControl-Steuererelement unterstützt die Konfigurierbarkeit der Registerkartenknöpfe-Positionierung, die optionale Durchführung von PostBacks zum Server bei Betätigung der Registerkartenknöpfe, sowie Mouse-Hover-Effekte beim Bewegen der Maus über eine Registerkarte. Neben der Möglichkeit clientseitige Ereignisse abzufangen sind jedoch keine weiteren nennenswerten Rich Client-Funktionalitäten festzuhalten.

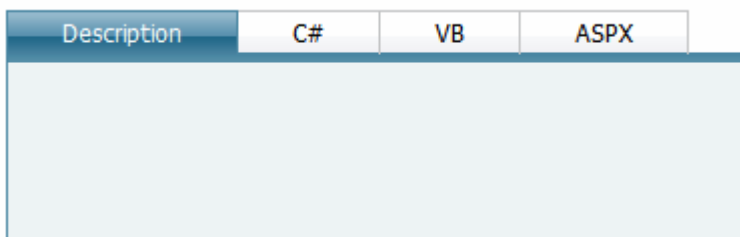


Abbildung 29: Das ASPxTabControl-Steuererelement

5.7.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Tab-Komponente wird über Request-Grenzen hinweg persistiert, falls JavaScript aktiviert ist. Bei deaktiviertem JavaScript ist auch kein Registerkartenwechsel ohne Weiteres

möglich, sodass die Betrachtung der Request-Lebenszyklus-Persistenz in diesem Fall zu verneinen ist.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können flexibel, auch über eigens dafür existierende Templates, angepasst werden.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist mit Ausnahme von Accesskeys nur äußerst eingeschränkt und nicht browserübergreifend möglich, da die Fokussierung der Tabknöpfe standardmäßig nur bei aktiviertem bzw. verfügbarem JavaScript funktioniert.

Bewertung: 0,5

Entwicklungsunterstützung

Es besteht eine umfangreiche, designergestützte Registerkartenentwicklung. Besonders hervorzuheben ist in diesem Zusammenhang der ASPxPageControl TabPage Editor, über welchen alle Tabs komfortabel aufgebaut werden können. Auch können Anpassungen an der Darstellung einzelner Registerkartenknöpfe hierüber erfolgen. Weitere Erleichterung wird durch die Unterstützung von Smart Tags erreicht.

Bewertung: 1,0

5.7.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Tab-Control seine Funktionalität ein. Es ist standardmäßig kein Tabwechsel mehr möglich.

Bewertung: 0,0

5.7.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS global oder auch registerkartenknopflokale Anpassungen an der Darstellung vorzunehmen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig auf Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und folglich negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente nicht mehr verwendet werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können jedoch in den meisten Browsern nicht fokussiert werden, falls JavaScript deaktiviert ist. Tastaturkürzel (Accesskeys) auf Tabknopfebene werden nicht standardmäßig unterstützt.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.7.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.7.3.3 Menüs

5.7.3.3.1 Beschreibung

Menüs können über das ASPxMenu-Steuerelement (vgl. Abbildung 30) realisiert werden. Die Anzeige der Untermenüs eines Menüpunktes erfolgt dabei standardmäßig bei Bewegung des Mauszeigers über den jeweiligen Menüpunkt. Es werden sowohl reine Textmenüpunkte als auch Menüpunkte mit Text und Icons unterstützt. Die Darstellung kann global aber auch auf jeder Ebene flexibel über CSS angepasst werden, wobei die Möglichkeit besteht, Darstellungseigenschaften an untergeordnete Elemente zu vererben. Auch Animationen und Hovereffekte sowie DataBinding werden unterstützt.

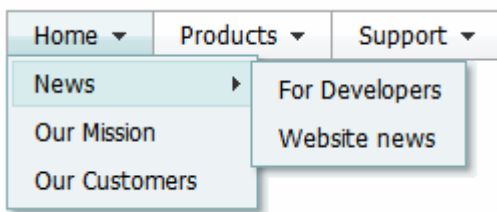


Abbildung 30: Das ASPxMenu-Control

5.7.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Es können verschachtelte Menüstrukturen erstellt werden. Neben einfachen Text-Menüpunkten können auch Symbole verwendet werden. Jeder Menüeintrag kann über eine Vielzahl von Einstellungen an die gewünschten Bedürfnisse adaptiert werden. Zudem können Darstellungseinstellungen an jeder Stelle einer Menüstrukturhierarchie an untergeordnete Menüelemente vererbt und dort gegebenenfalls wieder überschrieben werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine komfortable Bedienung mit der Tastatur ist nicht möglich. Eine Betätigung eines Menüelementes kann eine Navigation zu einer bestimmten Url oder die Ausführung einer serverseitigen Aktivität zur Folge haben.

Bewertung: 0,5

Entwicklungsunterstützung

Menükomponenten können über einen Visual Studio Designer, den ASPxMenu MenuItem Editor, komfortabel definiert werden. Daneben verfügt die Komponente über SmartTag-Unterstützung und diverse Steuerelementeigenschaften.

Bewertung: 1,0

5.7.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr angezeigt und keine PostBacks mehr ausgeführt.

Bewertung: 0,0

5.7.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale (über die Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften des jeweiligen Menüelements) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Menu-Komponenten basieren standardmäßig auf Html-Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und daher negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr auswählbar sind und PostBacks nicht mehr ausgeführt werden.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur ist nicht verfügbar.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.7.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.7.3.4 Symbolleisten

5.7.3.4.1 Beschreibung

Für Symbolleisten existiert keine dezidierte Komponente. Es lassen sich jedoch vertikale Symbolleistenstrukturen über das ASPxNavBar-Steuerelement (vgl. Abbildung 31) realisieren, weshalb im Folgenden dieses Steuerelement im Sinne einer Symbolleistenkomponente näher betrachtet werden soll.

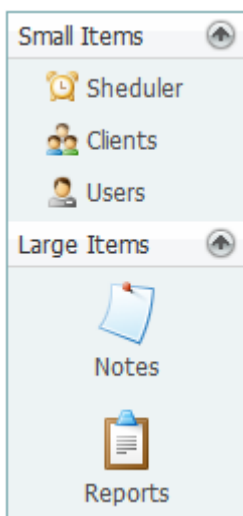


Abbildung 31: Das ASPxNavBar-Control

5.7.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Das ASPxNavBar-Control unterstützt die gruppierte Darstellung von Symbolen (und Texten). Es können die Gruppenüberschriften selbst, aber auch die einzelnen Symbolbereiche über Templates angepasst werden. Daneben können globale, gruppenglobale oder auch iconlokale Darstellungsanpassungen vorgenommen werden. Eine symbolleistentypisch horizontale Darstellung gestaltet sich jedoch äußerst problematisch.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers zeigt das entsprechende Icon im festgelegten Hover-Zustand an. Ein Klick darauf führt die hinterlegte Aktion aus bzw. erlaubt die Navigation zu einer hinterlegten Url. Eine komfortable Ansteuerung mit der Tastatur wird nicht unterstützt.

Bewertung: 0,5

Entwicklungsunterstützung

Symbolleistenkomponenten können über einen Visual Studio Designer, den ASPxNavBar Item Editor komfortabel bearbeitet werden. Zudem besteht SmartTag-Unterstützung.

Bewertung: 1,0

5.7.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität komplett ein. So werden keine serverseitigen Ereignisse ausgelöst und auch keine Hover-Effekte visualisiert.

Bewertung: 0,0

5.7.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

ASPxNavBar-Komponenten basieren auf Html-Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und folglich negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr auf Benutzereingaben reagiert.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird explizit nicht unterstützt.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur Verbesserung der Strukturierung und Navigation werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.7.3.4.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.7.3.5 Hierarchische Steuerelemente (Tree Views)

Es wird kein spezielles Steuerelement für hierarchische Strukturen mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines TreeView-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.7.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert. ASP .NET stellt jedoch eine Implementierung eines Fortschrittsanzeige-Steuerelements standardmäßig bereit, weshalb im Rahmen der Bewertung auf die ASP.NET-Implementierung verwiesen werden soll.

5.7.3.7 Infotips

5.7.3.7.1 Beschreibung

Zur Realisierung eines Infotips soll auf die bereits im Dialogkontext näher untersuchte Komponente ASPxPopupControl (vgl. Abbildung 32) näher eingegangen werden. Diese stellt unter Anderem die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element den vorgegebenen Inhalt einzublenden. Auch Animationen und Vergrößerung des Bereichs oder Verschieben des Bereichs wird zur Laufzeit unterstützt.

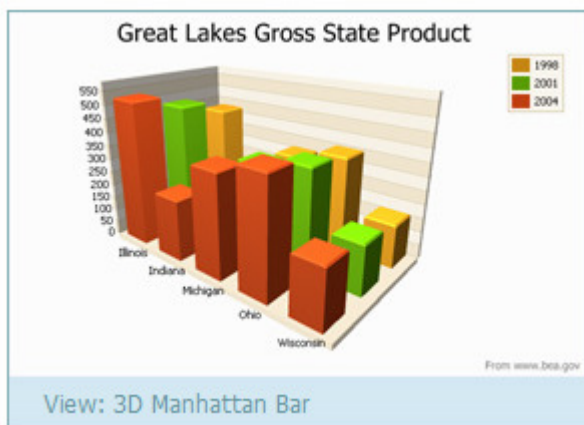


Abbildung 32: Die ASPxPopupControl-Komponente

5.7.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Die grundsätzliche graphische Darstellung (Rechteck mit Titel- bzw. Fußleiste; Anzeige konfigurierbar) ist vorgegeben. Es besteht jedoch umfassender Handlungsspielraum zur Anpassung der Darstellung via CSS und Templates.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf ein entsprechendes Element zeigt den entsprechenden Infotip-Inhalt bei entsprechender Konfiguration (auslösendes Element und Ereignis zur Anzeige) an. Das Intervall, nachdem der InfoTip und wie lange er angezeigt werden soll, ist frei konfigurierbar (AppearAfter- und DisappearAfter-Eigenschaften).

Bewertung: 1,0

Entwicklungsunterstützung

Über den ASP.NET-Standarddesigner können ASPxPopupControl-Steuerelement-Eigenschaften festgelegt werden. Zudem kann der Inhalt des Infotips via Drag & Drop befüllt werden. Auch besteht eine SmartTag-Unterstützung.

Bewertung: 1,0

5.7.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität ein. Das PopupControl wird dann nicht mehr angezeigt. Ein Ein- und Ausblenden ist dann auch nicht mehr möglich.

Bewertung: 0,0

5.7.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass der Dialograhmen bzw. Infotiprahmen flexibel angepasst werden kann. Die anzuzeigenden Inhalte können zudem frei festgelegt werden, sodass eine Zuständigkeit des Entwicklers hinsichtlich dieses Aspekts vorliegt. Sämtliche Komponentenbereiche lassen sich über CSS anpassen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die Komponente wird per Default über Tabellenelemente realisiert. Die innere Gestaltung obliegt dem Entwickler.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass kein Infotip beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Einblendung des Dialogs bzw. Infotips keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur unabhängig von der Verfügbarkeit von JavaScript ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog- bzw. InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch, wenngleich diese Möglichkeit optional besteht, um fremde Inhalte einzubinden. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.7.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.7.4 Fazit

DevExpress ASPxperience bietet viele nützliche, reichhaltige und flexible Komponenten, die die Produktpalette von ASP.NET nicht nur abrunden, sondern erheblich erweitern. Wenngleich viele Komfortfunktionen ohne großen Konfigurations- und Entwicklungsaufwand ab Werk bereitstehen, wird Aspekten aus dem Bereich Barrierefreiheit nur geringfügig Rechnung getragen, wobei als Beispiel die oftmals fehlende Möglichkeit Komponenten mit der Tastatur anzusteuern, genannt werden soll. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.8 JSF (Standard)

5.8.1 Beschreibung

JSF³⁸ (vgl. Geary und Horstmann, 2007) ist eine auf der Java EE-Plattform³⁹ basierende, komponentenorientierte Spezifikation von Sun zur Webentwicklung. Es liegt eine Referenzimplementierung von Sun vor. Daneben existieren weitere freie (z.B. MyFaces⁴⁰) und kommerzielle (z.B. Oracle ADF Rich Client) Implementierungen

Im Rahmen dieser Evaluierung sollen die Standard-Komponenten der JSF-Referenzimplementierung von Sun untersucht werden. Diese ist aufgrund der Standardisierung als repräsentativ für andere JSF-Frameworks, die die JSF-Spezifikation implementieren, zu sehen. Diese ist jedoch von darauf aufbauenden Frameworks, wie z.B. MyFaces Tomahawk zu unterscheiden, die Standard-JSF (realisiert durch MyFaces Core) um weitere Komponenten erweitern, wobei diese Frameworks dann in den folgenden Kapiteln näher untersucht werden.

Zu beachten ist noch, dass es aufgrund der Vielzahl an Entwicklungsumgebungen im Javaumfeld als schwierig zu bewerten ist, eine Entwicklungsumgebung als Standard zu betrachten. Aufgrund der großen Verbreitung soll jedoch bei entsprechenden Integrationsanforderungsprüfungen standardmäßig von Eclipse ausgegangen werden, außer beim entsprechenden Framework liegt explizit eine optimierte Integration mit einer anderen Entwicklungsumgebung vor bzw. es wird eine andere Entwicklungsumgebung empfohlen.

5.8.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 1.2_08 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.8.3 Detailuntersuchung

5.8.3.1 Dialoge

Es wird kein spezielles Dialog-Steuerelement mitausgeliefert. Dieses ist durch auf JSF aufbauende Frameworks bereitzustellen, weshalb eine Gesamtbewertung von 0,0 für JSF in diesem Aspekt vorzunehmen ist.

5.8.3.2 Registerkarten

Es werden keine Registerkartensteuerelemente mit JSF-RI ausgeliefert.

5.8.3.3 Menüs

Es werden keine Menüsteuerelemente mit JSF-RI ausgeliefert.

³⁸ Vgl. <http://www.ibm.com/developerworks/java/library/j-jsf3/>.

³⁹ Vgl. <http://java.sun.com/javaee/reference/>.

⁴⁰ Vgl. Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden..**

5.8.3.4 Symbolleisten

Es werden keine Symbolleistensteuerelemente mit JSF-RI ausgeliefert.

5.8.3.5 Hierarchische Steuerelemente (Tree Views)

Es werden keine Steuerelemente zur Darstellung hierarchischer Daten in Form von TreeViews mit JSF-RI ausgeliefert.

5.8.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mit JSF-RI ausgeliefert.

5.8.3.7 Infotips

Es wird kein spezielles Infotip-Steuerelement mit JSF-RI ausgeliefert.

5.8.4 Fazit

JSF-RI bietet eine stabile Basis für die Erstellung von Java-basierenden Webanwendungen; es fehlen aber Webkomponenten, die zur Erstellung reichhaltiger Web 2.0-Systeme erforderlich sind. Diese Bewertung muss jedoch nicht negativ interpretiert werden, da es auch als eine Architektur- bzw. Designentscheidung von JSF-RI gesehen werden kann, Rich Client-Frameworks nicht in die Basisbibliotheken zu integrieren, sondern stattdessen über Erweiterungs- bzw. Komponentenbibliotheken zur Verfügung zu stellen.

5.9 MyFaces Tomahawk

5.9.1 Beschreibung

Bei Apache MyFaces⁴¹ (vgl. Marinschek et al., 2006) handelt es sich um eine Open Source-Alternative zur JSF-Referenzimplementierung von Sun Microsystems. MyFaces gliedert sich in verschiedene Teilprojekte:

- Core JSF-1.1⁴²
- Core JSF-1.2⁴³
- Trinidad⁴⁴
- Tobago⁴⁵
- Tomahawk⁴⁶
- Sandbox⁴⁷
- Orchestra⁴⁸
- Portlet Bridge⁴⁹

⁴¹ Vgl. <http://www.myfaces.org/>.

⁴² Vgl. <http://myfaces.apache.org/core11/index.html>.

⁴³ Vgl. <http://myfaces.apache.org/core12/index.html>.

⁴⁴ Vgl. <http://myfaces.apache.org/trinidad/index.html>.

⁴⁵ Vgl. <http://myfaces.apache.org/tobago/index.html>.

⁴⁶ Vgl. <http://myfaces.apache.org/tomahawk/index.html>.

⁴⁷ Vgl. <http://myfaces.apache.org/sandbox/index.html>.

⁴⁸ Vgl. <http://myfaces.apache.org/orchestra/index.html>.

⁴⁹ Vgl. <http://myfaces.apache.org/portlet-bridge/index.html>.

Tomahawk stellt dabei erweiterte, über die im JSF-Standard beschriebene, hinausgehende Funktionalität, primär in Form von Steuerelementen, zur Verfügung. Dazu zählen beispielsweise Registerkarten- und Menüsteuerelemente. Trinidad bietet unter Anderem verbesserte JSF-Converter (mit clientseitiger Unterstützung) und Validatoren. Aufgrund des Fokus dieser Studie auf Rich Client-Komponenten soll jedoch im Folgenden MyFaces Tomahawk näher beleuchtet werden.

5.9.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist Version 1.1.6 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.9.3 Detailuntersuchung

5.9.3.1 Dialoge

5.9.3.1.1 Beschreibung

Tomahawk bietet keine Komponenten, die direkt für den Einsatzbereich Dialoge konzeptioniert worden sind. Jedoch kann die Komponente CollapsiblePanel (vgl. Abbildung 33) für diesen Einsatzbereich zweckentfremdet benutzt werden. Es werden jedoch keine Komfortfunktionen wie Drag & Drop-Verschiebung, Dialoggrößenänderung, Anzeige eines Dialogschattens sowie Laden anderer Webseiten in den Dialog out of the box unterstützt.

Bei Spezifikation keines Headerbereichs in der Komponente (header-Facet) wird wie in Abbildung 33 dargestellt, bei eingeblendetem Dialog ein „v“-Symbol angezeigt. Per Klick darauf erfolgt das Ausblenden des Dialogs und es wird ein „>“-Symbol angezeigt, wobei ein erneuter Klick dies wiederum umgekehrt. Die Anzeige bzw. das Verbergen des Pseudodialogs erfolgt serverseitig, der zugehörige serverseitige Aufruf jedoch per Default über JavaScript.

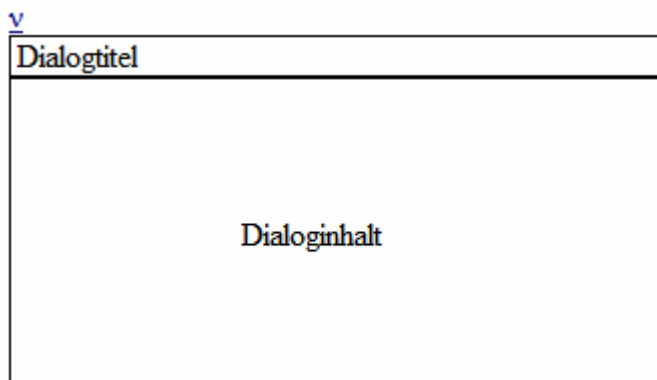


Abbildung 33: Das CollapsiblePanel-Steuerelement als Dialog mit Link-Trigger

5.9.3.1.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Dialog-Komponente wird standardmäßig über Requestgrenzen hinweg persistiert. Dies erfolgt jedoch nur bei aktiviertem JavaScript korrekt, da bei deaktiviertem bzw. nicht verfügbarem JavaScript kein Zustandswechsel (Ein- bzw. Ausblenden) mehr möglich ist. Es

existiert zudem kein dialogoriginärer Mechanismus zum Datenaustausch zwischen Dialog und der dialogöffnenden Umgebung (bis auf reguläres JSF-Binding).

Bewertung: 0,5

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Das Steuerelement erlaubt die Integration beliebiger anderer Komponenten. Die Darstellung lässt sich komplett über CSS anpassen, da keine Vorgaben gemacht werden. Komfortfunktionen wie Drag & Drop oder Verschiebung des Dialogs durch den Anwender stehen nicht zur Verfügung.

Bewertung: 0,5

Verhalten

Es gibt keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Über die Defaulteinstellung oder den Header-Facet-Bereich der Komponente ist es jedoch möglich entsprechende Toggle-Elemente festzulegen, wobei dann jedoch auf JSF-Standardmittel zurückgegriffen wird. Eine komfortable deklarative Festlegungsmöglichkeit besteht jedoch nicht. Standardtastenkürzel für häufige Dialogaktionen, wie das Schließen eines Dialogs, werden nicht ohne Entwicklungsaufwand unterstützt.

Bewertung: 0,0

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Eclipse und WTP⁵⁰ bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Auch können Komponenteneigenschaften darüber ohne Programmieraufwand festgelegt werden.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Inhalt eines Dialoges kann nicht ohne zusätzliche Komponenten bzw. Tags von einer externen Ressource geladen werden (keine dialogoriginäre Unterstützung). Es liegt zudem keine dialogoriginäre Implementierung zum kontrollierten Datenaustausch zwischen Dialoginhalt und dialogöffnender Umgebung vor.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

⁵⁰ Web Tools Platform (WTP); vgl. <http://www.eclipse.org/webtools/>.

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge ohne eigenen Entwicklungs- bzw. Anpassungsaufwand unterstützt.

Bewertung: 0,0

5.9.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein.

Bewertung: 0,0

5.9.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header-, Footer- und Content-Bereich sowie weitere Bereiche) eine Anpassung über CSS-Klassen bzw. Skins unterstützt wird, da alle Bereiche vom Entwickler selbst ohne Unterstützung durch die Komponente zu implementieren sind.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die äußere und innere Gestaltung des Dialogs obliegt dem Entwickler, weshalb keine als negativ zu bewertenden Vorgaben durch die Komponente vorliegen.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann nicht mehr über dialogoriginäre Mittel ein- bzw. ausgeblendet werden kann

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (z.B. Anzeigen, Verbergen) existiert jedoch keine explizite, dialogoriginäre Accesskey-Unterstützung.

Bewertung: 0,5

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuerelementen (CollapsiblePanel-Steuerelemente) um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.3.2 Registerkarten

5.9.3.2.1 Beschreibung

MyFaces Tomahawk stellt über das TabbedPane-Steuerelement (vgl. Abbildung 34) eine Komponente zur Realisierung von Registerkarten zur Verfügung. Dabei kann über diese Komponente eine client- oder auch serverseitige Umschaltung zwischen den angezeigten Registerkarten realisiert werden.

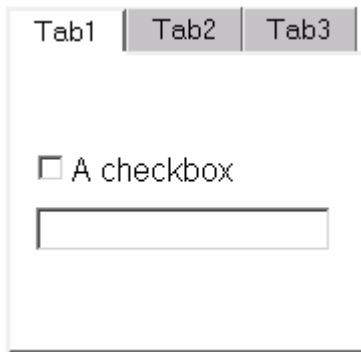


Abbildung 34: Das TabbedPane-Steuerelement

5.9.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Registerkarten-Komponente wird über Request-Grenzen hinweg persistiert. Auch bei deaktiviertem JavaScript ist ein Registerkartenwechsel ohne Weiteres möglich.

Bewertung: 1,0

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können jedoch nur einfach über CSS und nicht komplett frei, z.B. über Facets, angepasst werden.

Bewertung: 0,5

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist über die Tab-Taste möglich. Registerkartenabhängige Accesskeys werden jedoch nicht unterstützt.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Auch können Komponenteneigenschaften darüber ohne Programmieraufwand festgelegt werden.

Bewertung: 1,0

5.9.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript funktioniert das TabbedPane-Steuerelement aufgrund der dann serverseitigen Tabwechselimplementierung weiterhin ohne Einschränkung.

Bewertung: 1,0

5.9.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS global oder auch registerkartenknopflokale Anpassungen an der Darstellung vorzunehmen. Eine template- bzw. facetbasierte Anpassungsmöglichkeit hinsichtlich der Registerkartenknöpfe ist jedoch derzeit noch nicht gegeben.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig auf Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und folglich negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript ist ebenso ohne negative Folgen.

Bewertung: 1,0

Anforderung 7

Diese Anforderung ist im Kontext der Registerkartensteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können über die Tab-Taste fokussiert werden. Tastaturkürzel (Accesskeys) auf Tabknopfebene werden nicht standardmäßig unterstützt.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.3.3 Menüs

5.9.3.3.1 Beschreibung

Menüs können über das JSCookMenu-Steuerelement (vgl. Abbildung 35) oder alternativ über das PanelNavigation2-Steuerelement (vgl. Abbildung 36) realisiert werden.

Das JSCookMenu-Steuerelement basiert dabei auf der gleichnamigen, unabhängig von MyFaces Tomahawk erhältlichen JavaScript-Bibliothek JSCookMenu und abstrahiert die zugrundeliegenden JavaScript-Aufrufe weitgehend.

Das PanelNavigation2-Steuerelement ist der Nachfolger des PanelNavigation-Steuerelements und ermöglicht allgemein die Darstellung von horizontalen und vertikalen Navigationsstrukturen. Beide Komponenten lassen sich über CSS anpassen, wobei zudem Hover-Effekte für die einzelnen Items festlegbar sind.

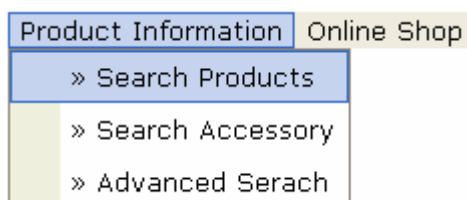


Abbildung 35: Das JSCookMenu-Control



Abbildung 36: Das PanelNavigation2-Steuerelement als Menü

5.9.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Über beide Komponenten können verschachtelte Menüstrukturen abgebildet werden. Neben einfachen Text-Menüpunkten können auch Symbole verwendet werden. Jeder Menüeintrag kann über Stylesheets angepasst werden. Bei JSCookMenu ist es zusätzlich möglich Themes zu benutzen. Es existieren jedoch bei beiden Komponenten keine ausgereiften templatebasierten Ansätze mit elementlokalen Attributzuweisungsoptionen und der Möglichkeit von Vererbung, wie es bei vorangehenden Lösungen bei zusätzlicher Designerunterstützung aufgezeigt worden ist.

Bewertung: 0,5

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende potentielle Untermenü an. Eine Bedienung mit der Tastatur ist beim PanelNavigation2-Steuerelement möglich, nicht jedoch beim JSCookMenu. Bei beiden Komponenten kann jedoch die dahinterliegende Funktion nur bei aktiviertem JavaScript ausgeführt werden. Die Action-Attribut-unabhängige Navigation zu einer anderen Ressource über t:commandNavigation2 und die Eigenschaft externalLink innerhalb einer PanelNavigation2-Komponente ist jedoch unabhängig von JavaScript stets möglich.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Der Aufbau der Menüs wird jedoch durch keinerlei IDE-Designer oder ähnliche Mechanismen im Besonderen unterstützt, sodass der Aufbau im Quellcode erfolgen wird.

Bewertung: 0,5

5.9.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control (bei beiden Komponenten) seine Funktionalität ein. Es werden keine Untermenüpunkte mehr dynamisch angezeigt und keine PostBacks mehr ausgeführt. Eine non-action-Attribut-basierte Navigation ist jedoch davon unabhängig weiterhin möglich.

Bewertung: 0,0

5.9.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen bei beiden Realisierungsmöglichkeiten umfangreiche Möglichkeiten um globale (über die Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften der untergeordneten einzelnen Komponenten) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Menu-Komponenten auf JSCookMenu-Basis basieren standardmäßig auf Html-Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und daher negativ zu bewerten ist. PanelNavigation2-Komponenten erlauben jedoch auch standardmäßig die Ausgabe als Html-Listenelemente (ul und li).

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr auswählbar sind und PostBacks nicht mehr ausgeführt werden. Im Fall von JSCookMenu wird bei Nicht-Verfügbarkeit von JavaScript das Menü selbst nicht mehr dargestellt. Eine non-action-Attribut-basierte Navigation ist jedoch bei Verwendung der Komponente PanelNavigation2 und einer untergeordneten CommandNavigation2-Komponente davon unabhängig weiterhin möglich.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur ist nur beim PanelNavigation2-Steuerelement möglich. Bei deaktiviertem JavaScript bringt dies jedoch keine Vorteile, da dann die zu den Navigationspunkten hinterlegten Funktionen (mit Ausnahme einfacher Navigation) nicht mehr aufrufbar sind.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung werden beim JSCookMenu keine besonderen Strukturierungselemente eingesetzt. Beim PanelNavigation2 werden hingegen Html-Listenelemente verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente, bis auf die Html-Listenelemente beim PanelNavigation2, eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.3.4 Symbolleisten

5.9.3.4.1 Beschreibung

Für Symbolleisten existiert keine dezidierte Komponente. Es lassen sich jedoch Symbolleistenstrukturen durch das PanelNavigation2-Steuerelement in Kombination mit untergeordneten CommandNavigation-Steuerelementen, welche wiederum GraphicImage-Steuerelemente enthalten, realisieren (vgl. Abbildung 37), weshalb im Folgenden das PanelNavigation2 Steuerelement im Sinne einer Symbolleistenkomponente näher betrachtet werden soll. Wenngleich sich auch mit JSCookMenu bei entsprechender Anpassung entsprechende Realisierungen vornehmen lassen, soll im Rahmen dieser Untersuchung der Fokus auf die nahe liegendere PanelNavigation-Komponente gesetzt werden, da diese im Rahmen der Untersuchung unter dem Aspekt Menüs einige Vorteile im Rahmen barrierefreier Komponenten, wie die Tastaturansteuerungsmöglichkeit über die Tab-Taste, bietet.



Abbildung 37: Das PanelNavigation2-Steuerelement als Symbolleiste

Dabei ist hervorzuheben, dass einer panelNavigation2-Komponente entweder über commandNavigation2 oder über NavigationMenuItem typisch aussehende Symbolleistenschaltflächen untergeordnet werden können.

5.9.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Neben einfachen Textsymbolleistenflächen können auch typische Symbole verwendet werden. Prinzipiell kann jede beliebige Komponente als Symbolleistenelement verwendet werden, aufgrund der Überlagerung von Link-Funktionalität und der Funktionalität der beinhalteten Komponente empfiehlt es sich jedoch, um zudem browserunabhängig zu bleiben, sich auf einfache Symbolleistenicons, gegebenenfalls mit Beschriftungen, zu beschränken. Jedes Symbolleistenelement kann über Stylesheets angepasst werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers zeigt das entsprechende Icon im festgelegten Hover-Zustand an. Ein Klick darauf führt die hinterlegte Aktion aus bzw. erlaubt die Navigation zu einer hinterlegten Url. Eine Ansteuerung mit der Tastatur wird über die Tab-Taste ermöglicht.

Symbolleistensteuerelementoriginäre Accesskey-Unterstützung besteht jedoch nicht, was aber durch die jeweiligen verwendeten Komponenten innerhalb des PanelNavigation2-Steuerlements realisiert werden kann. Bei deaktiviertem JavaScript ist es jedoch nicht mehr möglich die zu den Symbolleistenflächen hinterlegten Aktionen auszuführen (mit Ausnahme der non-action-Attribut basierten Navigation via untergeordneter CommandNavigation2-Komponente).

Bewertung: 0,5

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Der Aufbau der Symbolleisten wird jedoch durch keinerlei IDE-Designer oder ähnliche Mechanismen im Besonderen unterstützt, sodass der Aufbau im Quellcode erfolgen wird.

Bewertung: 0,5

5.9.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität komplett ein (mit Ausnahme der non-action-Attribut basierten Navigation via untergeordneter CommandNavigation2-Komponente). So werden keine serverseitigen Ereignisse ausgelöst und auch keine Hover-Effekte visualisiert.

Bewertung: 0,0

5.9.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistelementlokale CSS-Vorgaben anzuwenden. Die globalen Vorgaben können direkt Styleklassenangaben bei der PanelNavigation2-Komponenten vorgegeben werden. Die lokalen Vorgaben sind über die jeweils als Kindelemente von PanelNavigation2 angebotenen Styleklassen anpassbar.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die PanelNavigation2-Komponente erlaubt standardmäßiges Rendering als HTML-Listenelemente.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass diese nicht mehr auf Benutzereingaben reagiert (mit Ausnahme der non-action-Attribut basierten Navigation via untergeordneter CommandNavigation2-Komponente).

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird über die Tab-Taste unterstützt. Die Unterstützung von Accesskeys obliegt dem jeweiligen Entwickler, da dieser prinzipiell beliebige Steuerelemente innerhalb des PanelNavigation2-Steuerelements platzieren kann.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur Verbesserung der Strukturierung und Navigation werden Html-Listenelemente verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden Html-Listenelemente eingesetzt.

Bewertung: 1,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.4.5 Browserinteroperabilität

Das Symbolleisten-Steuerelement in Form der PanelNavigation2-Komponente funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.3.5 Hierarchische Steuerelemente (Tree Views)

5.9.3.5.1 Beschreibung

Hierarchische Strukturen können unter MyFaces Tomahawk über die Tree2-Komponente (vgl. Abbildung 38) realisiert werden. Diese erlaubt nicht die statische Spezifikation von untergeordneten Strukturen, aber dafür die Bindung an ein BackingBean wodurch dynamisch beliebige hierarchische Strukturen gebunden und über die Komponente visualisiert werden können. Nahezu jede JSF-Komponente kann dabei innerhalb eines Baumknotens gerendert werden, wobei zudem die Wahl zwischen clientseitiger oder serverseitiger Knotenexpansion bzw. Knotenkollabierung besteht.



Abbildung 38: Die Tree2-Komponente

5.9.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist. Bei deaktiviertem bzw. nicht verfügbarem JavaScript kann jedoch keine Expansion bzw. Kollabierung des Baumes mehr ausgeführt werden, sodass in diesem Fall auch von keiner echten Persistenz mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Das Tree2-Steuerelement ist flexibel anpassbar und ermöglicht die feingranulare Kontrolle über die Darstellung jedes einzelnen Knotens, basierend auf (frei definierbaren) Knotentypen. Dabei ist zu berücksichtigen, dass nahezu jede reguläre JSF-Komponente innerhalb eines Baumknotens untergebracht werden kann. So können neben Icons und Beschriftungen beispielsweise auch Checkboxes innerhalb von Baumknoten realisiert werden.

Bewertung: 1,0

Verhalten

Ein Anklicken eines Tree2-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses, sofern JavaScript aktiviert ist. Unterelemente werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch ebenso nur bei aktiviertem JavaScript.

Bewertung: 1,0

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Der Aufbau der Baumstrukturen wird jedoch durch keinerlei IDE-Designer oder ähnliche Mechanismen im Besonderen unterstützt, sodass der Aufbau im Quellcode (genauer gesagt im Backing Bean) erfolgen wird.

Bewertung: 0,5

5.9.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität ein. Ein Expandieren bzw. Kollabieren von Baumknoten ist dann ebenso wie die Verarbeitung serverseitiger Logik nicht mehr verfügbar, da serverseitige Funktionen via JavaScript aufgerufen werden.

Bewertung: 0,0

5.9.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können spezifische Alternativtexte über die jeweils innerhalb eines Baumknoten platzierten Komponenten vorgegeben werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global graphische Einstellungen vorgenommen werden können. Auch auf Baumknotenebene können DetailEinstellungen

hinsichtlich der Visualisierung, basierend auf den CSS-Klassen der in den Knoten platzierten Komponenten, vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet eine Zweckentfremdung des table-Elements bei dieser Komponente statt, da hierarchische Navigationsstrukturen standardmäßig über Tabellen abgebildet werden.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Knotenkollabierung bzw. Knotenexpansion und keine Ausführung von Befehlen mehr erfolgt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird eingeschränkt unterstützt, ist jedoch grundsätzlich von der Art der in den Baumknoten eingebetteten Komponenten abhängig.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.5.5 Browserinteroperabilität

Das Tree2-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.3.6 Fortschrittsanzeige

Es wird kein spezielles Fortschrittsanzeige-Steuerelement mitausgeliefert.

Bewertung: 0,0

5.9.3.7 Infotips

5.9.3.7.1 Beschreibung

Zur Realisierung eines Infotips soll auf die Tomahawk-Komponente Popup (vgl. Abbildung 39) näher eingegangen werden. Diese stellt unter Anderem die Funktion zur Verfügung, bei Bewegung des Mauszeigers über ein Element den vorgegebenen Inhalt einzublenden. Komfortfunktionen, wie beispielsweise Animationen werden jedoch nicht unterstützt.

textual text situation.

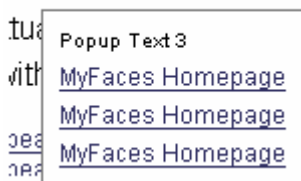


Abbildung 39: Die Tomahawk-Popup-Komponente

5.9.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Die Darstellung kann flexibel über CSS und ein der Komponente unterzuordnendes Facet namens popup angepasst werden. Einzig ein div-Element als Container für die spezifizierten Komponenten wird vorgegeben.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf ein entsprechendes Element zeigt den entsprechenden Infotip-Inhalt an. Weitere Einstellungen, wie z.B. das Interval, nachdem der InfoTip und wie lange er angezeigt werden soll, sind nicht konfigurierbar. Die Anzeige funktioniert zudem nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Eine komfortable Möglichkeit, über einen graphischen Designer neben der Komponente, zu der der InfoTip bei Mausbewegung darüber angezeigt werden soll, auch den anzuzeigenden Inhalt festzulegen existiert nicht. Die grundsätzlichen „flachen“ Eigenschaften lassen sich jedoch über Eclipse und WTP wie bei allen anderen JSF-Komponenten graphisch anpassen.

Bewertung: 0,5

5.9.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität ein. Bei Mausbewegungen über das entsprechende Trigger-Element wird dann kein Popup-Komponenteninhalte mehr angezeigt.

Bewertung: 0,0

5.9.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über JSF-Bordmittel zu realisieren, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die anzuzeigenden Inhalte zudem frei festgelegt werden können, sodass eine Zuständigkeit des Entwicklers hinsichtlich dieses Aspekts vorliegt. Sämtliche Komponentenbereiche lassen sich über CSS anpassen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die Komponente gibt lediglich das Containersteuerelement als div-Bereich vor. Der konkrete Inhalt kann durch den Entwickler, inklusive der CSS-Gestaltung des div-Bereichs, völlig frei vorgegeben werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was

bedeutet, dass kein Infotip beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Einblendung des Dialogs bzw. Infotips keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine spezielle Ansteuerung mit der Tastatur, unabhängig von der Verfügbarkeit von JavaScript, ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Dialog- bzw. InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch, wenngleich diese Möglichkeit optional besteht, um fremde Inhalte einzubinden. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.9.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.9.4 Fazit

MyFaces Tomahawk füllt die im Hinblick auf Web 2.0-Komponenten vorliegende Lücke der auf Basis der JSF-Spezifikation realisierten Basis-Frameworks wie JSF-RI oder MyFaces Core. Dabei bietet das MyFaces Tomahawk viele nützliche, reichhaltige und flexible Komponenten für nahezu alle postulierten erforderlichen Steuerelemente. Wie auch bei den vorangegangenen Frameworks wird Aspekten aus dem Bereich Barrierefreiheit jedoch nur geringfügig Rechnung getragen, wobei als Beispiel die bei manchen Komponenten fehlende Möglichkeit diese mit der Tastatur anzusteuern, genannt werden soll. Ein Ausschlusskriterium im eGovernment-Kontext ist jedoch, dass die Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.10 JBoss RichFaces

5.10.1 Beschreibung

Bei JBoss RichFaces⁵¹ handelt es sich nach JBoss.org (2008a) um eine reichhaltige Komponentenbibliothek für JSF und ein fortschrittliches Framework um eigene Business Anwendungen mit AJAX-Technologie anzureichern. Ein wichtiger Aspekt ist dabei die

⁵¹ Vgl. <http://www.jboss.org/jbossrichfaces/>.

Erreichung der Verbesserung der Benutzererfahrung bei Verwendung von auf diesem Framework basierenden Webapplikationen.

Nach ebd. beinhaltet JBoss RichFaces unter Anderem folgende Schlüsselfunktionen:

- Schaffung von Webanwendungen, die so interaktiv wie Desktopanwendungen sind, ohne die Erfordernis von JavaScript-Programmierung
- Gleichzeitige Nutzung der Vorzüge von JSF und AJAX
- Aufrüstung bestehender Applikationen um AJAX-Funktionalität
- Themefähige, große Anzahl an Komponenten
- Entwicklung eigener Komponenten mit integrierter AJAX-Funktionalität
- Automatisierte Testfunktionalität zur Erzeugung von Testfällen bereits während der Komponentenerstellung

Details zu den Komponenten selbst können dem offiziellen Entwicklerhandbuch, welches auch als Anwenderhandbuch bezeichnet wird, entnommen werden (vgl. JBoss.org, 2008b).

5.10.2 **Evaluierte Produktversion**

Zum Zeitpunkt dieser Untersuchung ist Version 3.2.1 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.10.3 **Detailuntersuchung**

5.10.3.1 **Dialoge**

5.10.3.1.1 *Beschreibung*

RichFaces unterstützt die Realisierung von Dialogen primär über das ModalPanel-Steuerelement (vgl. Abbildung 40), wenngleich auch andere Komponenten, wie das Panel-Steuerelement bei entsprechenden Anpassungsarbeiten prinzipiell für diesen Anwendungsbereich geeignet sind. Es werden Komfortfunktionen wie Dialogmodalität, Drag & Drop-Verschiebung, Dialoggrößenänderung und in der Transparenz konfigurierbare Schatten unterstützt (vgl. JBoss.org, 2008b: 474ff.).

Bei Anzeige des Dialogs werden Elemente im Hintergrund automatisch gesperrt. Das Anzeigen und Verbergen des Dialogs erfolgt standardmäßig via JavaScript (vgl. ebd.: 473).

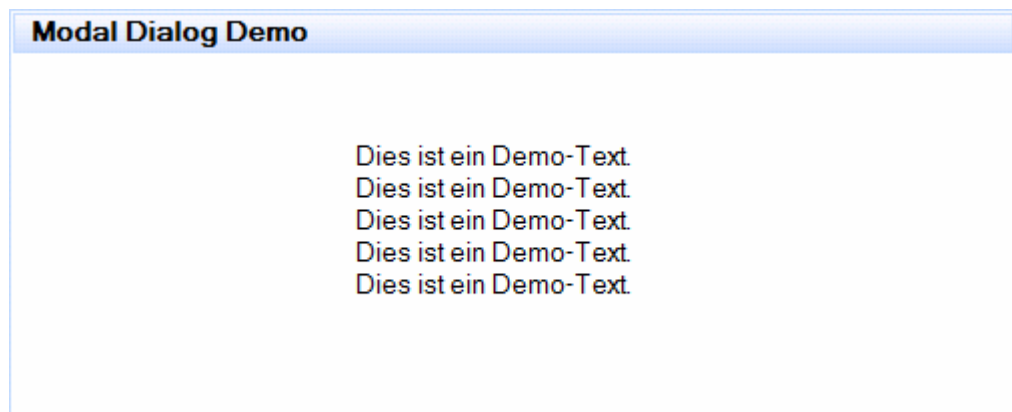


Abbildung 40: Das Rich Faces-ModalPanel-Steuerelement

5.10.3.1.2 *Rich-Client-Funktionalität*

Request-Lebenszyklus-Persistenz

Trotz Vorliegen einer Eigenschaft mit der die Persistierung aktiviert werden kann (keepVisualState; vgl. ebd.: 481), findet diese bei Stattfinden eines echten Non-Ajax-Submits nicht statt, da ein mit einem Accesskey versehener Button außerhalb des Dialogs platziert werden kann, welcher bei aktiviertem Dialog via Tastenkombination ausgelöst werden kann, was eine Darstellung des letzten serverseitig gerenderten Zustands zur Folge hat, aber durch JavaScript ausgelöste Visibilitätsänderungen ohne benutzerdefinierte Entwicklungstätigkeit nicht reflektiert.

Zudem erfolgt die Ein- und Ausblendung in der Standardimplementierung lediglich über JavaScript, sodass ohne JavaScript kein Zustandswechsel im Hinblick auf die Sichtbarkeit auftreten kann und damit auch eine Persistierung über Requestgrenzen hinweg hinfällig ist. Es existiert zudem kein dialogoriginärer Mechanismus zum Datenaustausch zwischen Dialog und der dialogöffnenden Umgebung (bis auf reguläres JSF-Binding).

Bewertung: 0,0

Anpassbarkeit

Der Dialog ist völlig frei anpassbar. Das Steuerelement erlaubt die Integration beliebiger anderer Komponenten. Die Darstellung lässt sich komplett über CSS anpassen, da keine Vorgaben gemacht werden. Komfortfunktionen wie Drag & Drop, Größenänderung oder Verschiebung des Dialogs durch den Anwender stehen zur Verfügung. Über Facets (header und controls, vgl. ebd.: 478f.) können der Headertitel angepasst und die anzuzeigenden Dialogoptionen über die Positionierung beliebiger Komponenten wie Schaltflächen, z.B. eine Schließen-Schaltfläche, angepasst werden.

Bewertung: 1,0

Verhalten

Es gibt keinerlei Unterstützung um komfortabel eine beliebige Anzahl von Komponenten als Öffnen- oder Schließen-Steuerelemente zu deklarieren. Jedem Element, welches ein Ein- oder Ausblenden des Dialogs bewirken soll, muss daher explizit JavaScript-Code oder ein JSF-Binding, welches den entsprechenden JavaScript-Code zur Laufzeit einfügt, zugewiesen werden (vgl. ebd.: 478). Bei nicht verfügbarem JavaScript existiert keine dialogoriginäre Möglichkeit den Dialog ein- oder auszublenden (ohne die Erfordernis entsprechende Entwicklungen vorzunehmen).

Bewertung: 0,0

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Auch können Komponenteneigenschaften darüber ohne Programmieraufwand festgelegt werden. Dem Dialog untergeordnete Komponenten können auch via Drag & Drop innerhalb des Dialogs platziert werden.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Inhalt eines Dialoges kann nicht ohne zusätzliche Komponenten bzw. Tags von einer externen Ressource geladen werden (keine dialogoriginäre Unterstützung). Es liegt zudem keine dialogoriginäre Implementierung zum kontrollierten Datenaustausch zwischen Dialoginhalt und dialogöffnender Umgebung vor.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge (z.B. Windows-ähnliche MessageBoxen) ohne eigenen Entwicklungs- bzw. Anpassungsaufwand unterstützt.

Bewertung: 0,0

5.10.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein.

Bewertung: 0,0

5.10.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Darstellung lässt sich völlig flexibel über Facets und die entsprechende Theming-Unterstützung anpassen.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header- und Inhaltsbereich) eine Anpassung über CSS-Klassen bzw. Skins unterstützt wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die äußere und innere Gestaltung des Dialogs obliegt dem Entwickler, weshalb keine als negativ zu bewertenden Vorgaben durch die Komponente vorliegen. Standardmäßig findet jedoch eine Zweckentfremdung des table-Elements bei dieser Komponente statt, da der Dialog standardmäßig über eine Html-Tabellenstruktur umgesetzt wird.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann nicht mehr über dialogoriginäre Mittel ein- bzw. ausgeblendet werden kann

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (z.B. Anzeigen, Verbergen), die ebenfalls über die Platzierung beliebiger Steuerelemente im controls-Facet-Bereich (Titelleistenbereich) realisiert werden können, existiert jedoch keine explizite, dialogoriginäre Accesskey-Unterstützung. Die Accesskeys können in diesem Fall jedoch über die platzierten Schaltflächen selbst vorgegeben werden.

Bewertung: 1,0

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den Dialog-Steuer-elementen (ModalPanel-Steuer-elemente) um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuer-elemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuer-elemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegt in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.2 Registerkarten

5.10.3.2.1 Beschreibung

RichFaces stellt über das TabPanel-Steuerelement (Container für alle Registerkarten; vgl. JBoss.org, 2008b: 631-640) und das Tab-Steuerelement (einzelne Registerkarte; vgl. ebd.: 641-652) eine Gesamtkomponente zur Realisierung von Registerkarten zur Verfügung (vgl. Abbildung 41). Dabei kann über diese Komponenten eine client- oder auch serverseitige Umschaltung zwischen den angezeigten Registerkarten realisiert werden.

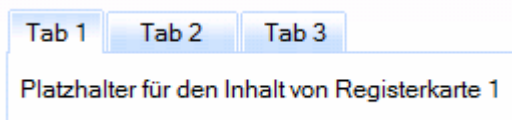


Abbildung 41: Das RichFaces-TabPanel-Steuerelement

5.10.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Registerkarten-Komponente wird über Request-Grenzen hinweg persistiert. Bei deaktiviertem JavaScript ist jedoch ein Registerkartenwechsel nicht mehr möglich, weshalb von keiner Persistierungsunterstützung in diesem Fall mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei anpassen. Die Registerkartenknöpfe können neben einfachen CSS-Anpassungen auch komplett frei über Facets, angepasst werden (vgl. ebd.: 646).

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte. Eine reine Bedienung mit der Tastatur ist nicht möglich, außer es erfolgt eine Anpassung der Registerkartentabs durch das label-Facet und dabei eine Spezifikation von Fokus- und Accesskey-akzeptierenden Komponenten.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Auch können Komponenteneigenschaften darüber ohne

Programmieraufwand festgelegt werden. Die einzelnen Registerkarten können über den angesprochenen Editor mit Drag & Drop befüllt werden.

Bewertung: 1,0

5.10.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript funktioniert das TabPanel-Steuerelement trotz der konfigurierbaren serverseitigen Tabwechselimplementierung (vgl. ebd.: 635f.) nicht mehr, da diese ebenfalls via JavaScript gesteuert wird.

Bewertung: 0,0

5.10.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die ASP.NET-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Darstellung ist zusätzlich über Themes und CSS anpassbar.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können auch flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS Anpassungen an der Darstellung vorzunehmen. Eine facetbasierte Anpassungsmöglichkeit hinsichtlich der Registerkartenknöpfe ist zudem gegeben.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig auf Tabellenelementen, was im Sinne der Barrierefreiheit als zweckentfremdend und folglich negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript führt unabhängig von der gewählten Tabwechselmoduseinstellung dazu, dass kein Tabwechsel mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Registerkartensteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können nicht über die Tab-Taste fokussiert werden. Auch Tastaturkürzel (Accesskeys) auf Tabknopfebene sind nicht standardmäßig vorgesehen. Über Facets könnten jedoch die Tabknopfbereiche frei angepasst werden, sodass sich entsprechende Funktionalität einfach nachrüsten lässt.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.2.5 Browserinteroperabilität

Das Registerkartensteuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.3 Menüs

5.10.3.3.1 Beschreibung

Menüs können über das RichFaces-DropDownMenu-Steuerelement (Abbildung 42) oder alternativ über das ContextMenu-Steuerelement (vgl. Abbildung 43) zur Abbildung von Kontextmenüs realisiert werden.

Im Folgenden soll der Fokus auf das RichFaces-DropDownMenu-Steuerelement und die zugehörigen unterzuordnenden Komponenten gerichtet werden, da Kontextmenüs nicht unter den zu untersuchenden Menüs aufgeführt werden (vgl. JBoss.org, 2008b: 349-3560).

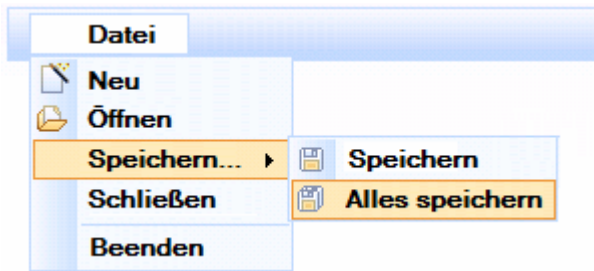


Abbildung 42: Das RichFaces-DropDownMenu-Control



Abbildung 43: Das RichFaces-ContextMenu-Control

5.10.3.3.2 Rich-Client-Funktionalität

Anpassbarkeit

Über die DropDownMenu-Komponente können verschachtelte Menüstrukturen abgebildet werden. Neben einfachen Text-Menüpunkten können auch Symbole verwendet werden. Jeder Menüeintrag kann über Stylesheets angepasst werden. Auch Themes werden unterstützt.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende Untermenü an. Eine Bedienung mit der Tastatur ist beim DropDownMenu-Steuerelement standardmäßig nicht möglich. Bei der Komponente kann ferner die dahinterliegende Funktion nur bei aktiviertem JavaScript ausgeführt werden.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Der Aufbau der Menüs wird jedoch durch keinerlei IDE-Designer oder ähnliche Mechanismen im Besonderen unterstützt, sodass der Aufbau im Quellcode oder über den Designer erfolgen wird, aber dabei nur über einfache Platzierung von neuen Menüelementen, ohne echtes WYSIWYG.

Bewertung: 0,5

5.10.3.3.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr dynamisch angezeigt, sodass auch die zugrundeliegenden Funktionen der entsprechenden Menüpunkte nicht mehr ausgeführt werden können.

Bewertung: 0,0

5.10.3.3.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Durch Theming und CSS bestehen ferner umfassende Anpassungsmöglichkeiten.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale (über die Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften der untergeordneten einzelnen Komponenten) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

DropDownMenu-Komponenten basieren standardmäßig auf Html-Div- und Html-Span-Elementen, was im Sinne der Barrierefreiheit als akzeptabel zu bewerten ist, wenngleich Html-Listenelemente hier die Struktur optimal zum Ausdruck bringen würden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr eingeblendet werden, wodurch eine Ausführung hinterlegter Funktionen auch nicht mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur ist standardmäßig nicht möglich.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung werden beim DropDownMenu keine besonderen Strukturierungselemente eingesetzt. Stattdessen finden Html-Span- und Html-Div-Elemente ihren Einsatz.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.3.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.4 Symbolleisten

5.10.3.4.1 Beschreibung

Für Symbolleisten kann das RichFaces ToolBar-Steuerelement (vgl. JBoss.org, 2008b: 667-673) eingesetzt werden. Dieses bietet nach ebd. unter Anderem Unterstützung von Gruppierung untergeordneter Komponenten (über die ToolBarGroup-Komponente; vgl. ebd.: 673-677), Separatoren, volle Skin-Unterstützung und die Möglichkeit beliebigen Inhalt aufzunehmen. Das RichFaces Toolbar-Steuerelement wird in Abbildung 44 dargestellt.



Abbildung 44: Das RichFaces-Toolbar-Steuerelement

5.10.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Dem ToolBar-Steuerelement kann beliebiger Inhalt untergeordnet werden. So können beispielsweise einfache Symbole, Symbole mit Text, Comboboxen oder Textfelder auf einer

ToolBar platziert werden. Jedes Symbolleistenelement kann über Stylesheets elementlokal oder auch global angepasst werden. Es gibt jedoch keine besondere Designerunterstützung, um Symbolleisten über graphische Werkzeuge zusammenzustellen. Dafür besteht jedoch die Möglichkeit via Drag & Drop Symbolleistenelemente zur Symbolleiste hinzuzufügen.

Bewertung: 1,0

Verhalten

Ein Klick auf ein Symbolleistenelement führt die hinterlegte Aktion aus bzw. erlaubt die Navigation zu einer hinterlegten Url. Eine Ansteuerung mit der Tastatur ist abhängig von den verwendeten, dem Symbolleistenelement untergeordneten, Komponenten. Symbolleistensteuerelementoriginäre Accesskey-Unterstützung besteht nicht, was aber durch die jeweiligen verwendeten Komponenten innerhalb des Steuerlements realisiert werden kann. Hover-Effekte werden zudem auch nicht dialogoriginär unterstützt, können aber auf Symbolleistenelementebene implementiert werden. Ob Nicht-Verfügbarkeit von JavaScript eine Auswirkung auf die Ausführung der beim jeweiligen Symbolleistenelement hinterlegten Anweisungen besitzt, hängt von den verwendeten Elementen ab und liegt somit in der Hand des Entwicklers.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Die Befüllung der Symbolleiste mit Komponenten kann damit auch via Drag & Drop erfolgen.

Bewertung: 1,0

5.10.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Die Komponente selbst erfordert keinerlei JavaScript. Konkrete JavaScript-Erfordernisse werden nur durch die untergeordneten Komponenten bestimmt.

Bewertung: 1,0

5.10.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden. Die globalen Vorgaben können direkt als Styleklassenangaben bei der ToolBar-Komponente vorgegeben werden. Die lokalen Vorgaben sind über die Styleklassen der jeweiligen, dem ToolBar-Steurelement untergeordneten, Komponenten anpassbar.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Das Rendering erfolgt standardmäßig als Tabelle, was im Sinne der Barrierefreiheit als Zweckentfremdung und damit negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript hat keine bestimmten Folgen auf die Symbolleistenkomponente. Die Auswirkungen hängen von der JavaScript-Abhängigkeit der verwendeten, der Symbolleistenkomponente untergeordneten, Steuerelemente ab.

Bewertung: 1,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur hängt von den verwendeten Komponenten ab und obliegt damit dem jeweiligen Entwickler, da dieser prinzipiell beliebige Steuerelemente innerhalb des ToolBar-Steuerelements platzieren kann.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Es werden keine Elemente zur Verbesserung der Strukturierung und Navigation verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden Html-Tabellen statt Html-Listenelementen eingesetzt, was eine Zweckentfremdung darstellt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.4.5 Browserinteroperabilität

Das Symbolleisten-Steuerelement in Form der PanelNavigation2-Komponente funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.5 Hierarchische Steuerelemente (Tree Views)

5.10.3.5.1 Beschreibung

Hierarchische Strukturen können unter RichFaces über die Tree-Komponente (vgl. JBoss.org, 2008b: 685-702) und unterzuordnende TreeNode-Steuerelemente (vgl. ebd.: 702-712) realisiert werden. Prinzipiell kann einem Baumknoten beliebiger Inhalt untergeordnet werden, wobei zudem Templates über Facets bereitgestellt werden, über welche die Darstellung eines Baumknotens zu bestimmten Zuständen angepasst werden kann. So werden die Facets iconCollapsed, iconExpanded, icon und iconLeaf unterstützt (vgl. ebd.: 708).

Zur Expandierung bzw. Kollabierung von Knoten werden die drei Modi Client, Server und Ajax unterstützt, wobei selbst die serverseitige Variante bei deaktiviertem JavaScript nicht funktioniert, da JavaScript zur Triggerung des serverseitigen Codes vorausgesetzt wird.

Ferner soll darauf hingewiesen werden, dass diese Komponente out of the box Drag & Drop-Unterstützung bereitstellt.

Die RichFaces Tree-Komponente wird in Abbildung 45 dargestellt.

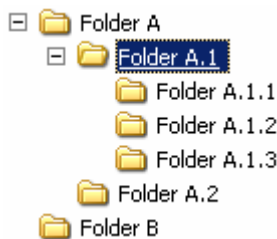


Abbildung 45: Die RichFaces-Tree-Komponente

Die deklarative Definition von Baumstrukturen bei gleichzeitiger Binding an Backing Bean-Datenquellen kann über die Komponenten TreeNodesAdapter bzw. RecursiveTreeNodesAdaptor (für rekursive Strukturen) erfolgen.

5.10.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist. Bei deaktiviertem bzw. nicht verfügbarem JavaScript ist (auch bei serverseitigem Modus) kein Expandieren bzw. Kollabieren der Baumknoten mehr möglich, sodass in diesem Fall auch von keiner echten Persistenz mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Das Tree-Steuerelement ist flexibel anpassbar und ermöglicht die feingranulare Kontrolle über die Darstellung jedes einzelnen Knotens, basierend auf den Baumknoten-Facets `iconCollapsed`, `iconExpanded`, `icon` und `iconLeaf` (vgl. ebd.: 708). Dabei ist zu berücksichtigen, dass nahezu jede reguläre JSF-Komponente innerhalb eines Baumknotens (`TreeNode`) untergebracht werden kann. So können neben Icons und Beschriftungen beispielsweise auch Checkboxes innerhalb von Baumknoten realisiert werden.

Bewertung: 1,0

Verhalten

Unterelemente eines Baumknotens werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch nur bei aktiviertem JavaScript. Ein Anklicken eines Tree-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses. Die Details hierzu, inklusive der JavaScript-Abhängigkeit, werden durch die jeweils verwendeten, untergeordneten Komponenten bestimmt.

Bewertung: 0,0

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Eclipse und WTP bereitgestellt. Dies schließt unter anderem einen JSF-Oberflächendesigner, den Web Page Editor, mit ein, wodurch eine Autovervollständigung für Expression Bindings und die Möglichkeit der Drag & Drop-Komponentenplatzierung besteht. Der Aufbau der Baumstrukturen wird durch den IDE-Designer dadurch unterstützt, dass der Komponentenbaum unterzuordnender Komponenten via Drag & Drop aufgebaut werden kann.

Bewertung: 1,0

5.10.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität dahingehend ein, dass kein Expandieren bzw. Kollabieren von Baumknoten dann mehr möglich ist (auch bei serverseitigem SwitchMode-Modus). Damit ist auch kein Zugriff auf untergeordnete Elemente mehr möglich, sodass auch die dahinterliegenden Funktionen nicht mehr nutzbar sind.

Bewertung: 0,0

5.10.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können spezifische Alternativtexte über die jeweils innerhalb eines Baumknoten platzierten Komponenten vorgegeben werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts

Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global graphische Einstellungen vorgenommen werden können. Auch auf Baumknotenebene können Detaileinstellungen hinsichtlich der Visualisierung, basierend auf den CSS-Klassen der in den Knoten platzierten Komponenten, vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet eine Zweckentfremdung des table-Elements bei dieser Komponente statt, da hierarchische Strukturen standardmäßig über Tabellen abgebildet werden.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Knotenkollabierung bzw. Knotenexpansion und damit keine Ausführung von ausgeblendeten, untergeordneten von Befehlen aufgrund der nicht mehr möglichen Einblendbarkeit mehr erfolgt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine

Ansteuerung mit der Tastatur ist grundsätzlich von der Art der in den Baumknoten eingebetteten Komponenten abhängig.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt, da standardmäßig auf Html-Tabellenlayouts gesetzt wird.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.5.5 Browserinteroperabilität

Das Tree2-Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.6 Fortschrittsanzeige

5.10.3.6.1 Beschreibung

Länger andauernde Vorgänge können über Fortschrittsanzeige-Steuerelemente visualisiert werden. RichFaces bietet hierzu das sogenannte ProgressBar-Steuerelement (vgl. JBoss.org, 2008b: 573-584) an, über welches der genaue Status eines Vorgangs dem Anwender mitgeteilt wird (vgl. Abbildung 46).

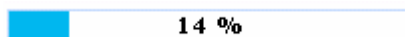


Abbildung 46: Das RichFaces-ProgressBar-Steuerelement

Alternativ dazu kann das Status-Steuerelement (vgl. ebd.: 130-133) eingesetzt werden, welches jedoch keinen genauen Fortschrittswert, sondern nur die Information, dass gerade ein Vorgang läuft, zurückliefert (vgl. Abbildung 47). Über die Facets start und stop kann dabei die Anzeige bei aktivem und inaktivem Vorgang frei angepasst werden.

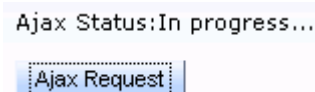


Abbildung 47: Exemplarischer Einsatz des RichFaces-Status-Steuerelements

Beide Elemente sich hinsichtlich der Gestaltung völlig anpassbar.

5.10.3.6.2 Rich-Client-Funktionalität

Anpassbarkeit

Sowohl das ProgressBar- als auch das Status-Steuerelement sind flexibel anpassbar. So kann der komplette anzuzeigende Inhalt, inklusive Animationen (beispielsweise über GIF-Dateien), beim Status-Steuerelement völlig frei für die Zustände „aktiver Vorgang“ und „inaktiver Vorgang“ über die Facets start und stop vorgegeben werden. Auch das ProgressBar-Steuerelement erlaubt die Anpassung der dargestellten Fortschrittsleiste durch die Spezifikation von CSS-Klassen.

Bewertung: 1,0

Verhalten

Bei Ausführung einer beliebigen AJAX-Anforderung wird automatisch das Status-Steuerelement aktiv, welches den definierten Inhalt während der Laufzeit des Vorgangs einblendet. Das Steuerelement wird jedoch naturgemäß nur bei AJAX-Requests tätig. Das ProgressBar-Steuerelement auf der anderen Seite muss explizit verwendet werden und bezieht den jeweiligen

Fortschrittswert von einer Backing Bean-Methode, die in regelmäßigen Zeitabschnitten gepollt wird.

Bewertung: 1,0

Entwicklungsunterstützung

Der Inhalt des Steuerelements kann sowohl über die Codeansicht als auch über den Designer vorgegeben werden.

Bewertung: 1,0

5.10.3.6.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt die Komponente ihre Funktionalität komplett ein.

Bewertung: 0,0

5.10.3.6.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können Alternativtexte spezifiziert werden. Dies liegt aber aufgrund der freien Anpassbarkeit vollständig in der Hand des Entwicklers.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass aufgrund der freien Anpassbarkeit der anzuzeigenden Inhalte durch den Entwickler, keine Bewertungsgrundlage hinsichtlich dieses Aspekts vorliegt, sodass wie beim vorherigen Punkt die maximale Leistungsstufe ohne Abzug vergeben wird.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die anzuzeigenden Inhalte werden komplett durch den Entwickler vorgegeben. Lediglich der Rahmen und der Fortschrittswert werden beim ProgressBar-Steuerelement durch die Komponente vorgegeben, wobei hier lediglich div- und span-Elemente verwendet werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin ohne Einschränkung benutzbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente ihre Funktion komplett einstellt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen. Jedoch kann die Visualisierung des Prozessfortschritts durch diese Komponente bereits als Unterstützung der Zeitvisualisierung betrachtet werden.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ergibt im Hinblick auf die Intention dieser Komponente keinen Sinn.

Bewertung: 1,0

Anforderung 10

Diese Komponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames eingesetzt. Die Strukturierung von Daten des Steuerelements obliegt aufgrund der freien Anpassbarkeit dem Entwickler.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen, wie z.B. die Bedeutung der Fortschrittskomponente, werden nicht automatisch bereitgestellt. Eine sinnvolle Gruppierung der beinhaltenden Inhalte obliegt dem Entwickler.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.6.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.3.7 Infotips

5.10.3.7.1 Beschreibung

Zur Realisierung eines Infotips soll auf die RichFaces-ToolTip-Komponente (vgl. JBoss.org, 2008b: 677-685) näher eingegangen werden (vgl. Abbildung 48). Diese stellt primär die Funktion zur Verfügung, bei einem bestimmten Ereignis, z.B. Bewegung des Mauszeigers über ein Element, den vorgegebenen Inhalt einzublenden. Dabei kann das Look & Feel angepasst werden und es werden auch verschiedene Arten, die anzuzeigenden Daten zu laden, unterstützt (vgl. ebd.: 677), was bedeutet, dass der anzuzeigende Inhalt zum Rendering-Zeitpunkt oder später via AJAX nachgeladen werden kann.

Dies ist ein Demotext.



Abbildung 48: Die RichFaces-ToolTip-Komponente

5.10.3.7.2 *Rich-Client-Funktionalität*

Anpassbarkeit

Die Darstellung kann flexibel über CSS angepasst werden. Die anzuzeigenden Inhalte und das Layout können völlig frei durch beliebige Steuerelemente, die dem ToolTip untergeordnet werden können, festgelegt werden. Für den Fall des dynamischen Nachladens via AJAX kann über das defaultContent-Facet zudem festgelegt werden, was während der Ladephase angezeigt werden soll.

Bewertung: 1,0

Verhalten

Bei Bewegung des Mauszeigers auf ein entsprechendes Element oder bei einem anderen frei konfigurierbaren clientseitigen Event wird der entsprechende Infotip-Inhalt angezeigt. Dabei kann das Look & Feel angepasst werden und es werden auch verschiedene Arten, die anzuzeigenden Daten zu laden, unterstützt (vgl. ebd.: 677), was bedeutet, dass der anzuzeigende Inhalt zum Rendering-Zeitpunkt oder später via AJAX nachgeladen werden kann. Die Einblendung des ToolTips funktioniert jedoch stets nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Eine komfortable Möglichkeit, über einen graphischen Designer neben der Komponente, zu der der InfoTip bei Mausbewegung darüber angezeigt werden soll, auch den anzuzeigenden Inhalt festzulegen existiert nicht. Die grundsätzlichen „flachen“ Eigenschaften lassen sich jedoch über Eclipse und WTP wie bei allen anderen JSF-Komponenten graphisch anpassen.

Bewertung: 0,5

5.10.3.7.3 *Funktionsweise bei nicht-verfügbarem JavaScript*

Bei deaktiviertem JavaScript stellt das Steuerelement seine Funktionalität ein. Bei Mausbewegungen über das entsprechende Trigger-Element oder anderen clientseitigen Ereignissen wird dann kein ToolTip-Inhalt mehr angezeigt.

Bewertung: 0,0

5.10.3.7.4 *Barrierefreiheit nach BITV*

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über JSF-

Bordmittel zu realisieren, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die anzuzeigenden Inhalte frei festgelegt werden können, sodass eine Zuständigkeit des Entwicklers hinsichtlich dieses Aspekts vorliegt.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Die Komponente gibt lediglich das Containersteuerelement als div-Bereich vor. Der konkrete Inhalt kann durch den Entwickler, inklusive der CSS-Gestaltung des div-Bereichs, völlig frei vorgegeben werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass kein Infotip beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Ein- und Ausblendung des Infotips (Eigenschaften showDelay und hideDelay) keine zeitgesteuerten Funktionalitäten vorliegen. Hinzu kommt, dass die angesprochenen Intervalle in der Regel äußerst kurz gehalten werden, da die Komponente nicht für lang andauernde Operationen vorgesehen ist, sondern zur zeitnahen Einblendung kontextsensitiver Hilfeinformationen oder weiterer Detailinformationen zu einem Element.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine spezielle Ansteuerung mit der Tastatur, unabhängig von der Verfügbarkeit von JavaScript, ist nicht vorgesehen.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch, wenngleich diese Möglichkeit optional besteht, um fremde Inhalte einzubinden. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.10.3.7.5 Browserinteroperabilität

Das jeweilige Steuerelement funktioniert unter allen spezifizierten Browsern.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.10.4 Fazit

JBoss RichFaces füllt ebenso wie MyFaces Tomahawk die im Hinblick auf Web 2.0-Komponenten vorliegende Lücke der auf Basis der JSF-Spezifikation realisierten Basis-Frameworks wie JSF-RI oder MyFaces Core. Dabei bietet das JBoss RichFaces im Vergleich zu MyFaces Tomahawk noch eine wesentlich größere Anzahl von Rich Client-Komponenten.

Im Kontext dieser Untersuchung liegt der Fokus jedoch auf einer definierten Menge an Komponenten und der Erfüllung von Aspekten unter Anderem aus dem Bereich Barrierefreiheit. Aspekten aus dem Bereich Barrierefreiheit wird jedoch nur geringfügig Rechnung getragen, wobei als Beispiel die bei manchen Komponenten fehlende Möglichkeit diese mit der Tastatur anzusteuern, genannt werden soll. Ein Ausschlusskriterium im eGovernment-Kontext ist zudem, dass nahezu alle Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.11 Oracle ADF Faces Rich Client

5.11.1 Beschreibung

Bei Oracle ADF Rich Client⁵², oder auch Oracle ADF RC bzw. ADF Faces, handelt es sich nach Oracle (2008: 1.1) um eine Sammlung von Standard-JSF-Komponenten⁵³ mit integrierter AJAX-Funktionalität.

Oracle ADF RC paart damit AJAX mit JSF-Technologien (vgl. ebd.: 1.3), wobei anhand des komponentenorientierten Ansatzes von JSF die Komplexität von AJAX reduziert und in wiederverwendbaren Komponenten gebündelt werden kann.

ADF Faces beinhaltet nach ebd. über 100 RIA-Komponenten, inklusive Dialogen, sortierbaren Tabellen und Baummenüs. Auch sind Komponenten zur Datenvisualisierung enthalten, wobei es sich um Flash- und SVG-fähige Komponenten handelt, die zur Darstellung von dynamischen

⁵²Vgl. <http://www.oracle.com/technology/products/adf/adffaces/index.html>.

⁵³Liste aller Komponenten: <http://jdevadf.oracle.com/adffacesdemo/faces/components/index.jspx>.

Diagrammen, Graphen und anderen Grafiken, welche auf Basis von zugrundeliegenden Daten in Echtzeit berechnet werden, eingesetzt werden können.

Nach Oracle (2008: 1.4) wird als auf ADF Faces optimal abgestimmte Entwicklungsumgebung Oracle JDeveloper angeboten, wodurch die Erstellung von JSF-Seiten über einen WYSIWYG-Editor einfach und schnell erfolgen kann.

5.11.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist das Paket JDeveloper 11g (mit integrierter ADF Faces-Version) aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

5.11.3 Detailuntersuchung

5.11.3.1 Dialoge

5.11.3.1.1 Beschreibung

ADF Faces unterstützt die Realisierung von Dialogen über die Steuerelemente `dialog` und `panelWindow` (vgl. Oracle, 2008: 12.12-12.18). Durch die Verwendung von bestimmten Präfix-Ausdrücken („`dialog`“) in den `faces-config.xml`-basierenden Navigationsregeln kann eine automatische Anzeige der entsprechenden Dialoge in einem neuen Browserfenster erfolgen.

Für diese Studie scheidet jedoch letztere genannter Aspekt aus, da damit „echte“ Popups erzeugt werden, was bedeutet, dass die jeweiligen Dialoginhalte in einem neuen Browserfenster angezeigt werden, wovon aus Gründen der Barrierefreiheit Abstand genommen werden soll. Dies ist auch bereits in der HTML Strict 1.1-Version ersichtlich, da dort das `target`-Attribut des Hyperlinkelements entfernt worden ist, um die Erzeugung von Popups auf Basis neuer Browserfenster zu verhindern (vgl. Pemberton, 2004).

Damit verbleibt für die weitere Untersuchung die Verwendung der angesprochenen Dialog-Komponenten als Kindelemente der sogenannten `popup`-Komponente, welche es ermöglicht, beliebige Elemente, unter Anderem damit auch Dialoge, dynamisch ein- und auszublenden. Wichtig ist hierbei, dass die entsprechenden Inhalte dann als Bestandteil der aktuellen Webseite und nicht als echtes Popup gerendert werden.

Die Komponenten `dialog` und `panelWindow` bewirken damit nur die Darstellung des Dialograhmens und des Inhalts, aber nicht das Verhalten der dynamischen Aus- und Einblendung, was auf generische Art und Weise durch die `popup`-Komponente als Containersteuerelement bereitgestellt wird (vgl. Oracle, 2008: 12.12-12.18). Aufgrund der großen Ähnlichkeit der `dialog`- und der `panelWindow`-Komponente soll im Folgenden stets die `dialog`-Komponente betrachtet werden, falls diese alleine bereits volle Anforderungserfüllung leistet.

Das ADF Faces-Dialog-Steuerelement wird in Abbildung 49, das ADF Faces-PanelWindow-Steuerelement in Abbildung 50 dargestellt.

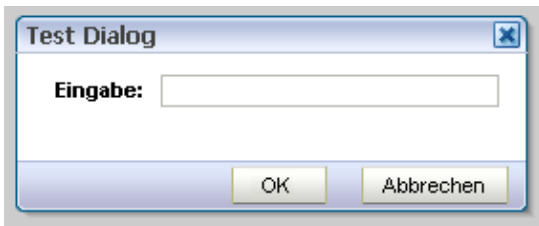


Abbildung 49: Das ADF Faces-Dialog-Steuererelement

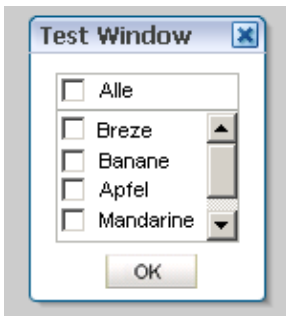


Abbildung 50: Das ADF Faces-PanelWindow-Steuererelement

5.11.3.1.2 *Rich-Client-Funktionalität*

Request-Lebenszyklus-Persistenz

Das Anzeigen bzw. Verbergen von Dialogen oder Fenstern über das popup-Steuererelement kann durch eine Kombination von ADF Faces-Command-Komponenten und untergeordneten ADF Faces-showPopupBehavior-Komponenten realisiert werden (vgl. Oracle, 2008: 12.16-12.18).

Hierbei erfolgt jedoch eine Umsetzung der Aufrufanweisungen über JavaScript-Code, sodass ein Anzeigen bzw. Verbergen eines Dialogs und damit auch eine requestübergreifende Persistierung von Zustandsänderungen in diesem Fall nicht möglich ist.

Aber auch bei aktiviertem JavaScript geht der Anzeigezustand bei Nicht-Ajax-Postbacks verloren. Dies kann über einen commandButton auf der jeweiligen, Dialog-öffnenden Webseite geprüft werden, indem dem jeweiligen Button außerhalb des Dialogs ein Accesskey zugewiesen wird, was es in der Teststellung ermöglicht, einen Submit trotz Dialogmodalität aufzurufen.

Es existiert zudem kein dialogoriginärer Mechanismus zum Datenaustausch zwischen Dialog und der dialogöffnenden Umgebung bei auf der jeweiligen Webseite eingebetteten Dialogen (bis auf reguläres JSF-Binding). Bei echten Popups, die nicht Gegenstand dieser Untersuchung sind, ist dies jedoch gegeben.

Bewertung: 0,0

Anpassbarkeit

Der Dialog stellt bereits eine Basisstruktur mit einer Titelleiste und einem optionalen Schließen-Button und einem frei vergebaren Titel bereit. Die Ausgabe kann über CSS angepasst werden. Da das Verhalten zur Anzeige von der popup-Komponente bereitgestellt wird, kann bei der Erfordernis weitreichender Anpassungen der Inhalt der popup-Komponente durch benutzerdefinierte bzw. andere Komponenten anstelle der Dialog-Komponente erstellt werden.

Bewertung: 1,0

Verhalten

Durch Unterordnung der ADF Faces-showPopupBehavior-Komponente können ADF Faces-command-Komponenten als Trigger zur Einblendung eines Dialogs deklariert werden. Es fehlt jedoch die Möglichkeit der deklarativen Festlegung einer beliebigen Anzahl an Komponenten zum Schließen des Dialog-popup-Steurelements. Bei nicht verfügbarem JavaScript existiert keine dialogoriginäre Möglichkeit den Dialog ein- oder auszublenden (ohne die Erfordernis entsprechende Entwicklungen vorzunehmen).

Bewertung: 0,0

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Oracle JDeveloper 11g bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner sowie Designer und Editoren für die verschiedenen JSF-typischen Entitäten, wie beispielsweise die Datei faces-config.xml mit ein. Auch können im Speziellen die Komponenteneigenschaften des Dialogs und seiner untergeordneten Komponenten darüber komfortabel festgelegt werden, was auch die Befüllung des Dialogs via Drag & Drop miteinschließt.

Bewertung: 1,0

Wiederverwendbarkeit von Dialoginhalten

Der Inhalt eines Inline-Dialoges (kein neues Browser-Fenster) kann nicht ohne zusätzliche Komponenten bzw. Tags von einer externen Ressource geladen werden (keine dialogoriginäre Unterstützung). Es liegt zudem keine dialogoriginäre Implementierung zum kontrollierten Datenaustausch zwischen Dialoginhalt und dialogöffnender Umgebung in diesem Fall vor.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre und dafür vorgesehene Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen, falls es sich um Dialoge ohne neues Browser-Fenster handelt. In diesem Fall sind Standard-JSF-Binding-Mechanismen zu benutzen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge (z.B. Windows-ähnliche MessageBoxen), aber Standard-Buttons für die Schaffung von Standarddialogen bereitgestellt, wodurch lediglich der entsprechende Text samt Symbol selbst implementiert werden muss, ergänzt durch die Ermittlung des jeweiligen Rückgabewertes.

Bewertung: 0,5

5.11.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein.

Bewertung: 0,0

5.11.3.1.4 *Barrierefreiheit nach BITV*

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole, wenn man vom Dialog schließen-Symbol absieht, welches jedoch ausgetauscht werden kann. Falls welche eingesetzt werden, sind diese über die JSF- bzw. ADF Faces-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Darstellung lässt sich völlig flexibel über CSS und alternativ Ersetzung der dialog-Komponente durch beliebige andere innerhalb des popup-Steuerelements anpassen.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header- und Inhaltsbereich) eine Anpassung über CSS-Klassen unterstützt wird. Der Inhaltsbereich kann dabei völlig frei angepasst werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die äußere und innere Gestaltung des Dialogs obliegt dem Entwickler, weshalb keine als negativ zu bewertenden Vorgaben durch die Komponente vorliegen. Standardmäßig findet jedoch eine Zweckentfremdung des table-Elements bei dieser Komponente statt, da der Dialog standardmäßig über eine Html-Tabellenstruktur umgesetzt wird, was sich jedoch durch Austausch der dialog-Komponente durch andere Komponenten innerhalb der popup-Komponente ändern lässt.

Bewertung: 0,5

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann nicht mehr über dialogoriginäre Mittel ein- bzw. ausgeblendet werden kann

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (Anzeigen, Verbergen) existiert jedoch keine explizite, dialogoriginäre Accesskey-Unterstützung. Die Accesskeys können in diesem Fall jedoch über manuell platzierte Schaltflächen vorgegeben werden.

Bewertung: 1,0

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich bei den dialog- bzw. popup-Steuerelementen um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von

zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.1.5 Browserinteroperabilität

Die Dialog-Komponenten funktionieren nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

5.11.3.2 Registerkarten

5.11.3.2.1 Beschreibung

ADF Faces unterstützt die Realisierung von Registerkarten über das Steuerelement `panelTabbed` (vgl. Abbildung 51) zur Registerkartenkomponentendarstellung in Kombination mit `showDetailItem`-Komponenten zur Darstellung der einzelnen Registerkarten (vgl. Oracle, 2008: 7.34-7.43).

Eine `showDetailItem`-Komponente stellt dabei lediglich einen Container für eine Registerkarte dar, bestimmt aber nicht die Darstellung, wodurch `showDetailItem`-Komponenten auch anderen Steuerelementen, wie z.B. dem `panelAccordion` (vgl. Abbildung 52), untergeordnet werden können, woraus eine komplett andere Darstellung resultiert.

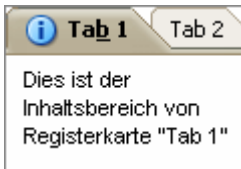


Abbildung 51: Das ADF Faces-PanelTabbed-Steuerelement



Abbildung 52: Das ADF Faces-PanelAccordion-Steuerelement

5.11.3.2.2 *Rich-Client-Funktionalität*

Request-Lebenszyklus-Persistenz

Der Zustand der Registerkarten-Komponente wird über Request-Grenzen hinweg persistiert. Bei deaktiviertem JavaScript ist jedoch ein Registerkartenwechsel nicht mehr möglich, weshalb von keiner Persistierungsunterstützung in diesem Fall mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei durch die Platzierung und Konfiguration beliebiger JSF-Komponenten (innerhalb einer showDetailItem-Containerkomponente) anpassen. Die Registerkartenknöpfe können via CSS angepasst werden. Eine templatebasierte Anpassbarkeit besteht nicht; wird jedoch auch nicht vorausgesetzt.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige der zugehörigen Registerkarte (repräsentiert durch eine showDetailItem-Komponente). Eine reine Bedienung mit der Tastatur ist über Accesskeys oder über die Tab-Taste möglich. Ein Anklicken einer Registerkarte oder eine Bedienung mit der Tastatur, z.B. über Accesskeys, zeigt jedoch nur bei aktiviertem JavaScript einen Effekt, da ohne JavaScript die zugrundeliegenden Funktionen nicht ausgeführt werden.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Oracle JDeveloper 11g bereitgestellt. Im Kontext von Registerkartenkomponenten können dabei neue Tabs über eine Art SmartTag bzw. Kontextmenü auf der Oberfläche der jeweiligen Registerkartenkomponente hinzugefügt werden. Die Tabs selbst können via Drag & Drop mit beliebigen JSF-Komponenten befüllt werden.

Bewertung: 1,0

5.11.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript funktioniert das TabPanel-Steuerelement nicht mehr, da der Tabwechsel JavaScript erfordert.

Bewertung: 0,0

5.11.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die JSF-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Darstellung ist zusätzlich über CSS anpassbar.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS Anpassungen an der Darstellung vorzunehmen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig auf Div-Elementen, was im Sinne der Barrierefreiheit als zweckmäßig und folglich positiv zu bewerten ist, da keine bei anderen Lösungen häufig vorzufindende Tabellenentfremdung vorliegt.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript führt unabhängig von der gewählten Tabwechselmoduseinstellung dazu, dass kein Tabwechsel mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Registerkartensteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können über die Tab-Taste fokussiert werden. Auch Tastaturkürzel (Accesskeys) auf Tabknopfebene werden unterstützt. Jedoch sind sowohl die Tastenkürzel als auch Accesskeys nur sinnvoll nutzbar, wenn JavaScript aktiviert ist, da nur dann die zugeordneten Funktionen ausgeführt werden.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Tab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.2.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

5.11.3.3 Menüs

5.11.3.3.1 Beschreibung

Menüs können über die ADF Faces-Komponenten MenuBar, Menu und CommandMenuItem abgebildet werden (vgl. Oracle, 2008: 13.1-13.7). MenuBar dient dabei zur Erstellung einer Menüzeile, Menu zur Darstellung von Menüs mit Untermenüs bzw. Untermenüpunkten und CommandMenuItem zur Darstellung von Menüpunkten ohne weitere Menüs bzw. Menüpunkte (vgl. Abbildung 53).

Es werden Komfortfunktionen unterstützt, wie beispielsweise das Ablösen von Untermenüs vom Menü und die freie Positionierung des jeweiligen Menüs auf der Weboberfläche. Neben reinen Text-Menüpunkten werden auch Icons unterstützt. Zudem werden auch Accesskeys für die Menüeinträge unterstützt. Daneben soll darauf hingewiesen werden, dass Radio- und Checkboxenmenüpunkte unterstützt werden.

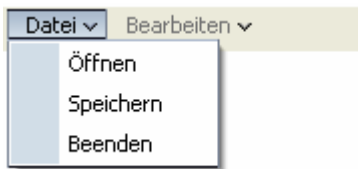


Abbildung 53: Die ADF Faces-Menükomponenten

5.11.3.3.2 *Rich-Client-Funktionalität*

Anpassbarkeit

Durch Kombination von MenuBar-, Menu- und CommandMenuItem-Komponenten können beliebig tiefe Menüstrukturen abgebildet werden, da Menu-Komponenten eine beliebige Anzahl an Menu- und CommandMenuItem-Komponenten enthalten können. Jedes CommandMenuItem kann dabei hinsichtlich Beschriftung, Icons, Shortcuts (Accesskeys) und der CSS-Darstellung bzw. dem Rendering als normales oder Radio- bzw. Checkboxmenü angepasst werden.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende Untermenü an. Eine Bedienung mit der Tastatur ist über die festlegbaren Accesskeys oder auch über die Tab- und Cursortasten möglich. Dies funktioniert jedoch alles nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine umfangreiche Entwicklungsunterstützung durch Oracle JDeveloper 11g bereitgestellt. Im Umfeld der Menükomponenten ist die Konfiguration der Menüelemente auf oberster Ebene hinsichtlich der Eigenschaften möglich. Untergeordnete Elemente müssen jedoch über die Code- bzw. Markupansicht manuell festgelegt werden.

Bewertung: 0,5

5.11.3.3.3 *Funktionsweise bei nicht-verfügbarem JavaScript*

Bei deaktiviertem JavaScript stellt das Menü-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr dynamisch angezeigt, sodass auch die zugrundeliegenden Funktionen der entsprechenden Menüpunkte nicht mehr ausgeführt werden können.

Bewertung: 0,0

5.11.3.3.4 *Barrierefreiheit nach BITV*

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Durch CSS bestehen ferner umfassende Anpassungsmöglichkeiten.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale (über die CSS-Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften der untergeordneten einzelnen Komponenten) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

ADF Faces-Menü-Komponenten basieren standardmäßig auf Html-Tabellenstrukturen, was im Sinne der Barrierefreiheit als negativ zu bewerten ist, da es eine semantische Zweckentfremdung darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr eingeblendet werden, wodurch eine Ausführung hinterlegter Funktionen auch nicht mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur ist nur bei aktiviertem JavaScript möglich.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung werden beim DropDownMenu keine besonderen Strukturierungselemente eingesetzt, da es sich standardmäßig um ein Tabellenlayout handelt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.3.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt offiziell keine Browserunterstützung vor. Durchgeführte Tests zeigten jedoch keine Probleme bei der Nutzung.

Bewertung: 0,5

5.11.3.4 Symbolleisten

5.11.3.4.1 Beschreibung

Für Symbolleisten kann das ADF Faces toolbar-Steuerelement in Kombination mit der commandToolBarButton-Komponente (vgl. Abbildung 54) eingesetzt werden (vgl. Oracle, 2008: 13.1-13.2 / Oracle, 2008: 13.7-13.13). Neben commandToolBarButton-Steuerelementen können aber prinzipiell auch beliebige andere Komponenten in der toolbar platziert werden, wie beispielsweise ein ADF Faces-inputText-Steuerelement.

Mehrere Toolbars können in einer Toolbox (Komponenten toolbox) zusammengefasst werden. Symbolleistenschaltflächen können Text, Icons oder eine Kombination davon darstellen. Mehrere toolbar-Komponenten können über die Komponenten ADF Faces-group gruppiert werden.

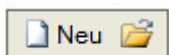


Abbildung 54: Das ADF Faces-Toolbar-Steuerelement in einer Toolbox-Komponente

5.11.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Dem toolbar-Steuerelement kann prinzipiell beliebiger JSF-Inhalt untergeordnet werden. Im Standardfall ist es jedoch vorgesehen, reguläre commandToolBarButton-Komponenten einzusetzen. Diese erlauben die Anzeige von Symbolleistenschaltflächen mit Text oder Icon oder einer Kombination davon. Der Symbolleistenbuttontyp kann über das type-Attribut angepasst werden, wodurch neben der regulären Ausgabe als Button auch das Rendering als Checkbox- oder Radiobutton im Stil der Symbolleiste unterstützt wird. Darüberhinaus werden auch Popups unterstützt. Die Darstellung lässt sich über CSS auf globaler und toolbarbuttonlokaler Ebene steuern.

Bewertung: 1,0

Verhalten

Ein Klick auf ein Symbolleistenelement führt die hinterlegte Aktion aus bzw. erlaubt die Navigation zu einer hinterlegten Url. Eine Ansteuerung mit der Tastatur ist über die Tab-Taste oder alternativ über Accesskeys möglich. Hover-Effekte werden zudem auch nicht

steuerelementoriginär unterstützt. Die Komponente lässt sich jedoch nur bei aktiviertem JavaScript benutzen, da nur dann hinterlegte Aktionen ausgeführt werden.

Bewertung: 0,5

Entwicklungsunterstützung

Eine Entwicklungsunterstützung durch JDeveloper 11g besteht. Im Kontext der Symbolleisten-erstellung umfasst diese die Drag & Drop basierte Zusammenstellung der Symbolleisteninhalte und die Festlegung der Eigenschaften der Symbolleiste sowie der untergeordneten Icons.

Bewertung: 1,0

5.11.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Die Komponente selbst funktioniert nur bei aktiviertem JavaScript korrekt. Falls jedoch entsprechende JavaScript-unabhängige Komponenten der Symbolleiste untergeordnet werden, kann dieses Defizit umgangen werden.

Bewertung: 0,5

5.11.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden. Die globalen Vorgaben können direkt als Styleklassenangaben bei der ToolBar-Komponente vorgegeben werden. Die lokalen Vorgaben sind über die Styleklassen der jeweiligen, dem Symbolleisten-Steuerelement untergeordneten, Komponenten anpassbar.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von

Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Das Rendering erfolgt standardmäßig als Tabelle, was im Sinne der Barrierefreiheit als Zweckentfremdung und damit negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript hat zur Folge, dass die herkömmlichen commandToolBarButton-Steuerelemente nicht mehr funktionieren. Dies kann jedoch durch die Verwendung anderer, JavaScript-unabhängiger JSF-Komponenten umgangen werden.

Bewertung: 0,5

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur hängt grundsätzlich von den verwendeten Komponenten ab. Bei commandToolBarButtons ist diese gegeben; jedoch nur bei aktiviertem JavaScript verwendbar, da ohne keine Ausführung der zugrundeliegenden Funktionen möglich ist.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Es werden keine Elemente zur Verbesserung der Strukturierung und Navigation verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden Html-Tabellen statt Html-Listenelementen eingesetzt, was eine Zweckentfremdung darstellt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.4.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt offiziell keine Browserunterstützung vor. Durchgeführte Tests zeigten jedoch keine Probleme bei der Nutzung.

Bewertung: 0,5

5.11.3.5 Hierarchische Steuerelemente (Tree Views)

5.11.3.5.1 Beschreibung

Hierarchische Strukturen können unter ADF Faces über die tree- (vgl. Oracle, 2008: 9.26-9.32) und die treeTable-Komponenten (vgl. Oracle, 2008: 9.32-9.34) abgebildet werden. Die treeTable-

Komponente vereint dabei die Charakteristika der beiden Komponenten tree und table. Klassische Bäume bzw. TreeViews werden über die Komponenten tree bereitgestellt, weshalb im Rahmen dieser Untersuchung diese näher betrachtet werden soll.

Die Datenbefüllung erfolgt über die Bindung an eine Backing Bean-Methode. Über das Facet nodeStamp kann das grundsätzliche Layout bzw. die Darstellung eines Knoten innerhalb des tree festgelegt werden. Über das Facet pathStamp kann die Darstellung hierarchischer Pfade angepasst werden.

Für eine exemplarische Darstellung der Komponente soll auf Abbildung 55 verwiesen werden.

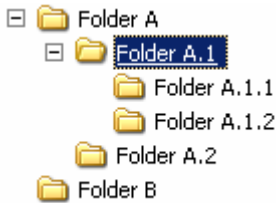


Abbildung 55: Die ADF Faces-Tree-Komponente

5.11.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist. Bei deaktiviertem bzw. nicht verfügbarem JavaScript wird der Tree überhaupt nicht mehr gerendert, sodass in diesem Fall auch von keiner echten Persistenz mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Das Tree-Steuerelement ist flexibel anpassbar. Zum Einen kann der Komponente eine CSS-Klasse zugewiesen werden, über welche prinzipiell auch untergeordnete Elemente graphisch angepasst werden können. Zum Anderen kann durch den Einsatz von Facets (nodeStamp und pathStamp) sowohl die Darstellung eines Knotens als auch der hierarchischen Pfadinformation angepasst werden.

Bewertung: 1,0

Verhalten

Unterelemente eines Baumknotens werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch nur bei aktiviertem JavaScript. Ein Anklicken eines Tree-Knotens bewirkt eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses. Die Details hierzu, inklusive der JavaScript-Abhängigkeit, werden durch die jeweils verwendeten, untergeordneten Komponenten bestimmt.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird Entwicklungsunterstützung durch Oracle JDeveloper 11g bereitgestellt. Im Kontext der Baumsteuerelemente ist neben der Festlegung von Komponenteneigenschaften über die GUI die Bearbeitung der unterstützten Facets über die Oberfläche zu hervorzuheben.

Bewertung: 1,0

5.11.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität dahingehend ein, dass kein Expandieren bzw. Kollabieren von Baumknoten mehr möglich ist.

Bewertung: 0,0

5.11.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können spezifische Alternativtexte über die jeweils innerhalb eines Baumknoten platzierten Komponenten vorgegeben werden.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global graphische Einstellungen über die Komponenten-CSS-Klasse vorgenommen werden können. Auch auf Baumknotenebene können über die Facets Detailinstellungen hinsichtlich der Visualisierung vorgenommen werden, was selbst die Festlegung beliebiger Knoteninhalte gestattet.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet keine Zweckentfremdung des table-Elements bei dieser Komponente statt, da hierarchische Strukturen standardmäßig über verschachtelte div- und span-Elemente aufgebaut werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Knotenkollabierung bzw. Knotenexpansion aufgrund fehlender Darstellung mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ist grundsätzlich von der Art der in den Baumknoten eingebetteten Komponenten abhängig. Grundsätzlich ist diese aber über die Cursortasten bzw. die Tab-Taste möglich.

Bewertung: 1,0

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt, da standardmäßig auf Html-Div- und Html-Span-Elemente gesetzt wird.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.5.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt keine Browserunterstützung vor. Tests zeigten, dass die initiale Darstellung erfolgreich ist, aber eine Kollabierung bzw. Expansion der Baumknoten scheitert.

Bewertung: 0,0

5.11.3.6 Fortschrittsanzeige

5.11.3.6.1 Beschreibung

Wenngleich eine Visualisierung des Fortschritts länger andauernder Vorgänge prinzipiell über beliebige ADF Faces Data Visualization-Komponenten (vgl. Oracle, 2008: 21.1-26.9) erfolgen kann, soll im Rahmen dieser Untersuchung das progressIndicator-Steuerelement (vgl. Abbildung 56) betrachtet werden, da dessen originäre Aufgabe die Visualisierung von Prozessfortschrittszuständen ist, sodass hier keinerlei zusätzlicher Programmieraufwand und optimale Anforderungserfüllung zu erwarten ist. Das Steuerelement kann in Kombination mit einem poll-Steuerelement verwendet werden um in zyklischen Abständen eine Aktualisierung des

Fortschrittszustands anzustoßen. Der aktuelle Wert kann über die value-Eigenschaft gebunden werden. Daneben wird über das an die value-Eigenschaft gebundene Objekt auch der Minimal- und Maximalwert zugeordnet.



Abbildung 56: Das ADF Faces-ProgressIndicator-Steuerelement

5.11.3.6.2 *Rich-Client-Funktionalität*

Anpassbarkeit

Das progressIndicator-Steuerelement ist über CSS anpassbar. Komplette templatebasierte Freiheit hinsichtlich anzuzeigender Farbverläufe bei der Fortschrittsleiste oder weitergehende Einstellungen lassen sich jedoch nicht deklarativ und ohne Programmieraufwand umsetzen.

Bewertung: 0,5

Verhalten

Bei Ausführung einer AJAX-Anforderung, resultierend aus der Triggerung durch eine Komponente, die als PartialTrigger im progressIndicator eingestellt wird, kann der progressIndicator aktualisiert werden. Dabei kann dem progressIndicator nicht nur ein einfacher Button sondern auch ein poll-Element zugeordnet werden, um periodische Aktualisierungen abzubilden. Naturgemäß stellt das Steuerelement bei deaktiviertem JavaScript seine Funktion ein, was aufgrund der Art dieser Komponente legitim ist.

Bewertung: 1,0

Entwicklungsunterstützung

Der Eigenschaften des Steuerelements können unter Oracle JDeveloper 11g sowohl über die Codeansicht als auch über den Designer (Property Inspector) vorgegeben werden.

Bewertung: 1,0

5.11.3.6.3 *Funktionsweise bei nicht-verfügbarem JavaScript*

Bei deaktiviertem JavaScript stellt die Komponente ihre Funktionalität komplett ein.

Bewertung: 0,0

5.11.3.6.4 *Barrierefreiheit nach BITV*

Anforderung 1

Keine der Komponenten gibt standardmäßig Audio- oder Videoinformationen aus.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts

Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass keine einschränkende Funktionalitäten im Hinblick auf diesen Aspekt durch die Komponente vorliegen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Für die Darstellung werden standardmäßig span-Elemente verwendet, was keine Zweckentfremdung von Tabellenstrukturen bedeutet.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin ohne Einschränkung benutzbar, da der aktuelle Fortschrittswert neben der graphischen Darstellung auch textuell ausgegeben werden kann. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente ihre Funktion komplett einstellt.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da in der Regel keine besonderen und vor allem langfristigen zeitgesteuerten Funktionalitäten im Rahmen eines einzelnen Prozessschritts vorliegen. Jedoch kann die Visualisierung des Prozessfortschritts durch diese Komponente bereits als Unterstützung der Zeitvisualisierung betrachtet werden.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur ergibt im Hinblick auf die Intention dieser Komponente keinen Sinn.

Bewertung: 1,0

Anforderung 10

Diese Komponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames eingesetzt. Die Strukturierung von Daten des Steuerelements obliegt aufgrund der freien Anpassbarkeit dem Entwickler.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen, wie z.B. die Bedeutung der Fortschrittskomponente, werden nicht automatisch bereitgestellt. Eine sinnvolle Gruppierung der beinhaltenden Inhalte obliegt dem Entwickler.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.6.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponenten gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

5.11.3.7 Infotips

5.11.3.7.1 Beschreibung

Zur Realisierung eines Infotips soll auf die ADF Faces-popup-Komponente (vgl. Abbildung 57) näher eingegangen werden (vgl. Oracle, 2008: 12.12-12.18). Diese bereits im Kontext der Dialoge betrachtete Komponente erlaubt die dynamische Ein- und Ausblendung beliebiger Inhalte und damit auch von Hilfeinformationen vielfältiger Natur. Dies umfasst die Darstellung von einfachen Texten, über Symbole bis hin zu komplexen Oberflächen mit Schaltflächen. Aufgrund der freien Festlegbarkeit der Inhalte kann auch die Darstellung über CSS flexibel gestaltet werden.

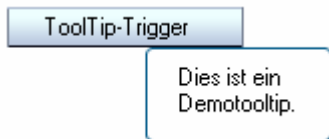


Abbildung 57: Die ADF Faces-Popup-Komponente als Tooltip

5.11.3.7.2 Rich-Client-Funktionalität

Anpassbarkeit

Die Darstellung kann flexibel über CSS angepasst werden. Die anzuzeigenden Inhalte können völlig frei durch beliebige JSF-Komponenten, die der popup-Komponente untergeordnet werden, bestimmt werden.

Bewertung: 1,0

Verhalten

Bei Bewegung des Mauszeigers auf ein entsprechendes Element oder bei einem anderen konfigurierbaren clientseitigen Event wird der entsprechende popup-Inhalt angezeigt. Das Triggering kann dabei über die jeweiligen als Auslöser-Komponenten zu deklarierenden Steuerelementen untergeordnete showPopupBehavior-Komponenten abgebildet werden. Das auslösende Ereignis lässt sich über das Attribut triggerType der showPopupBehavior festlegen. Hierbei werden die Einstellungen „action“ (Default-Werte; Mausklick bzw. Betätigung einer buttonähnlichen Komponente mit der Tastatur), „contextMenu“ (reguläre Aktion zum Einblenden eines Kontextmenüs) und „mouseHover“ unterstützt, wobei letztere Einstellung die für Tooltips und Infotips relevante darstellt, da damit bei Bewegung des Mauszeigers über einer Komponente die Einblendung des zugehörigen popup-Steuerelements erfolgen kann. Bei deaktiviertem JavaScript ist diese Funktionalität jedoch nicht verfügbar.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung durch Oracle JDeveloper 11g bereitgestellt. Dies schließt unter Anderem einen JSF-Oberflächendesigner sowie Designer und Editoren für die verschiedenen JSF-typischen Entitäten, wie beispielsweise die Datei faces-config.xml mit ein.

Auch können im Speziellen die Komponenteneigenschaften des Popups, d.h. InfoTips im konkreten Anwendungsszenarios, und seiner untergeordneten Komponenten darüber komfortabel festgelegt werden, was auch die Befüllung des Popups via Drag & Drop miteinschließt.

Bewertung: 1,0

5.11.3.7.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein.

Bewertung: 0,0

5.11.3.7.4 Barrierefreiheit nach BITV

Anforderung 1

Die Komponente gibt keine Audio- oder Videoinformationen aus. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet. Die Spezifikation von Bildern liegt aufgrund der völlig freien Befüllung des popup-Steuerelement-Inhaltsbereichs in der Verantwortung des Entwicklers und damit außerhalb der Komponente selbst.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass die anzuzeigenden Inhalte frei festgelegt werden können, sodass eine Zuständigkeit des Entwicklers hinsichtlich dieses Aspekts vorliegt.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Der Komponenteninhalt kann völlig frei und ohne Tabellenelemente festgelegt werden, wodurch sich Zweckentfremdungen (von Tabellenstrukturen) im Rahmen der Realisierung der Komponente ausschließen lassen.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass kein Infotip beim Bewegen des Mauszeigers auf ein Element mehr angezeigt werden kann.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht sinnvoll anwendbar, da bis auf die Zeit bis zur Ein- und Ausblendung des Infotips keine zeitgesteuerten Funktionalitäten vorliegen. Hinzu kommt, dass die angesprochenen Intervalle in der Regel äußerst kurz gehalten werden, da die Komponente nicht für lang andauernde Operationen vorgesehen ist, sondern im geprüften Anwendungsszenario zur zeitnahen Einblendung kontextsensitiver Hilfeinformationen oder weiterer Detailinformationen zu einem Element.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine spezielle Ansteuerung mit der Tastatur, unabhängig von der Verfügbarkeit von JavaScript, ist nicht explizit vorgesehen, wäre jedoch denkbar.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine echten Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die InfoTip-Steuerelemente selbst machen von keinerlei Frames Gebrauch, wenngleich diese Möglichkeit optional besteht, um fremde Inhalte einzubinden. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation sind nicht standardmäßig vorgesehen.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von diesen Komponenten zu implementieren.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.11.3.7.5 Browserinteroperabilität

Die Komponente funktioniert nach offizieller Aussage lediglich unter Internet Explorer 7+ und Firefox 2.0.0.2+.

Internet Explorer 6.0+

Erst ab Internet Explorer 7.0 wird die ordnungsgemäße Funktion der Komponente gewährleistet.

Bewertung: 0,5

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

Safari 3.1+

Es liegt keine Browserunterstützung vor.

Bewertung: 0,0

5.11.4 Fazit

Oracle ADF Faces stellt die bisher im Java EE-Bereich umfangreichste Webapplikationsbibliothek dar, wenngleich der Funktionsumfang selbst mit dem von JBoss RichFaces vergleichbar ist.

Wenngleich eine gute Anforderungserfüllung festzustellen ist, ist die sehr eingeschränkte Browserunterstützung als ein primäres Defizit zu nennen, da gegenwärtig nur ab Internet Explorer 7 und Firefox 2.0.0.2 eine vollständige Unterstützung der jeweiligen Funktionen zugesagt wird. Ein absolutes Ausschlusskriterium im eGovernment-Kontext ist zudem, dass nahezu alle Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

5.12 NetAdvantage for JSF

5.12.1 Beschreibung

Bei NetAdvantage for JSF⁵⁴ handelt es sich nach Infragistics (2008b: 2) um eine umfassende Sammlung von AJAX-aktivierten JSF-Komponenten zur Erstellung von Benutzeroberflächen gehobener Klasse auf Basis von J2EE-Applikationen. Dabei bietet NetAdvantage for JSF Komponenten wie hierarchische Grid-Komponenten, Kalender, Menüs, Dialoge, Eingabelemente und Registerkarten bei gleichzeitiger AJAX-Unterstützung und integrierten Visualisierungsstilen sowie umfangreicher Anpassungsmöglichkeiten.

Ferner besteht nach ebd. eine umfangreiche IDE-Unterstützung, sodass unter Anderem Eclipse 3.0, IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt werden.

Wenngleich die im Rahmen von NetAdvantage for JSF bereitgestellten Komponenten nach eigener Aussage Section 508⁵⁵ des Rehabilitation Act erfüllen (vgl. Infragistics, 2008b: 117), ist dies für diese Bewertung unerheblich, da Grundlage dieser Untersuchung die Anforderungen der BITV sowie darüber hinaus gehende Anforderungen des modernen Software Engineering sind.

5.12.2 Evaluierte Produktversion

Zum Zeitpunkt dieser Untersuchung ist die Version 2008 Volume 1 aktuell, weshalb sich die Untersuchung auf diese Version bezieht.

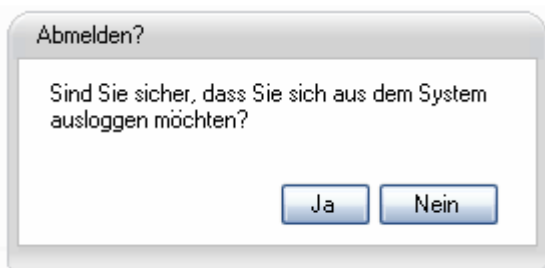
5.12.3 Detailuntersuchung

5.12.3.1 Dialoge

5.12.3.1.1 Beschreibung

NetAdvantage for JSF erlaubt die Realisierung von Dialogen über die WebDialogWindow-Komponente (vgl. Infragistics, 2008b: 237-250); dargestellt in Abbildung 58.

Die Dialog-Komponente wird dabei nach ebd. als div-Bereich gerendert, sodass keine Probleme mit regulären Popupblockern zu erwarten sind, da kein echtes Popup erzeugt wird. Als besondere Funktionalitäten werden modale und nicht-modale Dialoge, Titelzeilendarstellung, Größenänderung durch den Benutzer sowie Spezifikation des Ortes der Darstellung und des Fensterzustandes (minimiert, maximiert, verborgen) unterstützt.



⁵⁴ Vgl. <http://www.infragistics.com/java/netadvantage/jsf.aspx#Overview>.

⁵⁵ Vgl. <http://www.section508.gov/>.

5.12.3.1.2 *Rich-Client-Funktionalität*

Request-Lebenszyklus-Persistenz

Das Anzeigen bzw. Verbergen von Dialogen bzw. Fenstern erfolgt durch clientseitige Anpassung des WindowState des Dialogs. So sieht ein typisches Szenario vor, dass der jeweilige Dialog den WindowState hidden als initiale Einstellung besitzt; dieser aber über die set_windowState-Methode des Dialogs geändert werden kann. Unabhängig von JavaScript wird standardmäßig keine Persistierung des Anzeigezustands über Requestgrenzen hinweg vorgenommen, was bei AJAX-Requests keinerlei größere Nachteile besitzt, aber bei echten Submits einen kompletten Reset des Komponentenzustands bewirkt und somit echte Submits damit nicht mehr sinnvoll eingesetzt werden können.

Bewertung: 0,0

Anpassbarkeit

Der Dialog unterstützt eine Basisstruktur mit einer Titelleiste und optional einem Schließen-, Minimieren- und Maximieren-Button sowie einem frei vergebaren Titel bereit. Die Spezifikation der genannten Buttons erfolgt über eigene Komponenten dwMinimizeBox, dwMaximizeBox, dwCloseBox innerhalb der dwHeader-Komponente. Der Header kann aber statt diesen Standardkomponenten auch völlig frei mit beliebigen anderen Komponenten ausgestattet werden. Die Ausgabe kann zudem über CSS angepasst werden. Auch der Dialoginhalt kann templatebasiert angepasst werden. Dies wird durch die dwContentPane-Komponente realisiert, die die Platzierung beliebiger Subkomponenten gestattet.

Bewertung: 1,0

Verhalten

Das Einblenden und Ausblenden des Dialogs erfolgt via JavaScript. Es fehlt die Möglichkeit der deklarativen Festlegung einer beliebigen Anzahl an Komponenten zum Öffnen und Schließen des Dialog-Steuerelements. Bei nicht verfügbarem JavaScript existiert keine dialogoriginäre Möglichkeit den Dialog ein- oder auszublenden (ohne die Erfordernis entsprechende Entwicklungen vorzunehmen).

Bewertung: 0,0

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung unter Eclipse 3.0 (mit WTP), IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt. Dabei können im die Dialogelemente im Speziellen via Drag & Drop positioniert und ihre Eigenschaften festgelegt werden. Untergeordnete Steuerelemente lassen sich jedoch im Header- und Body-Bereich nur schwierig positionieren, sodass eine codebasierte Entwicklung nicht zu vermeiden ist, da der Designer keine WYSIWYG-Ansicht des Dialogs bietet.

Bewertung: 0,5

Wiederverwendbarkeit von Dialoginhalten

Der Inhalt eines Dialoges (kein neues Browser-Fenster) kann nicht ohne zusätzliche Komponenten bzw. Tags von einer externen Ressource geladen werden (keine dialogoriginäre Unterstützung).

Es liegt zudem keine dialogoriginäre Implementierung zum kontrollierten Datenaustausch zwischen Dialoginhalt und dialogöffnender Umgebung in diesem Fall vor.

Bewertung: 0,0

Definierter Datenaustauschprozess

Es existiert keine definierte, dialogimplementierungsoriginäre und dafür vorgesehene Möglichkeit um Informationen zwischen dem Dialog-Owner, also der öffnenden Webseite und dem Dialog auszutauschen, falls es sich um Dialoge ohne neues Browser-Fenster handelt. In diesem Fall sind Standard-JSF-Binding-Mechanismen oder JavaScript-Aufrufe zu benutzen.

Bewertung: 0,0

Standard-Dialoge für MessageBoxen

Es werden keine Standarddialoge (z.B. Windows-ähnliche MessageBoxen) für die Schaffung von Standarddialogen bereitgestellt.

Bewertung: 0,0

5.12.3.1.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt der Dialog seine Funktionen, wie das Ein- und Ausblenden, ein.

Bewertung: 0,0

5.12.3.1.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole, wenn man vom Dialog schließen-Symbol absieht, welches jedoch ausgetauscht werden kann. Falls welche eingesetzt werden, sind diese über die JSF- bzw. ADF Faces-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die entsprechenden Buttons zur Dialogmaximierung, Dialogminimierung und zum Schließen des Dialogs werden bereitgestellt. Der Header kann aber auch völlig frei festgelegt werden. Die Darstellung lässt sich zudem völlig flexibel über CSS anpassen.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für alle relevanten Bestandteile des Dialogs (Header- und Inhaltsbereich) eine Anpassung über CSS-Klassen und untergeordnete Komponenten (im Sinne von Templates) unterstützt wird. Der Inhaltsbereich kann dabei völlig frei angepasst werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Dialog-Komponenten vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von den Dialog-Steuerelementen zu sehen.

Bewertung: 1,0

Anforderung 5

Die Dialogdarstellung erfolgt standardmäßig über ein Tabellenlayout, was eine Zweckentfremdung darstellt und im Sinne der Barrierefreiheit als negativ zu betrachten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin verwendbar. Eine Deaktivierung von JavaScript bewirkt jedoch, dass die Komponente nicht mehr sinnvoll verwendet werden kann, da der Dialog dann nicht mehr über dialogoriginäre Mittel ein- bzw. ausgeblendet werden kann

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Dialogsteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt Analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine schlüssige Tabulatorreihenfolge ist standardmäßig gegeben. Tastaturkürzel (Accesskeys) können unabhängig vom Dialog für die enthaltenen Kindelemente vergeben werden. Für Dialogaktionen (Anzeigen, Verbergen) existiert jedoch keine explizite, dialogoriginäre, anpassbare Accesskey-Unterstützung. Die Accesskeys können in diesem Fall jedoch über manuell platzierte Schaltflächen vorgegeben werden.

Bewertung: 1,0

Anforderung 10

Es werden keine regulären Popups verwendet. Stattdessen handelt es sich um Bereiche der jeweiligen Webseite, die dynamisch ein- und ausgeblendet werden.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der Dialog-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Es werden keine Frames verwendet.

Bewertung: 1,0

Anforderung 13

Anforderung 13 ist im Kontext der Dialog-Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Eine sinnvolle Gruppierung von zusammengehörigen (Navigations-Elementen), wie Hyperlinks ist als unabhängig von der Dialog-Komponente zu betrachten.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte der Dialog-Komponenten liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.12.3.1.5 Browserinteroperabilität

Die Browserinteroperabilität stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.12.3.2 Registerkarten

5.12.3.2.1 Beschreibung

NetAdvantage for JSF ermöglicht die Realisierung von Registerkarten über das Steuerelement WebTab (vgl. Infragistics, 2008b: 345-350), welches aus den Komponenten tabView und tabItem besteht (vgl. ebd.: 345), wobei ersteres den umfassenden Registerkartencontainer und letzteres für die einzelnen Registerkarten verwendet wird. Eine exemplarische Darstellung der WebTab-Komponente kann Abbildung 59 entnommen werden.

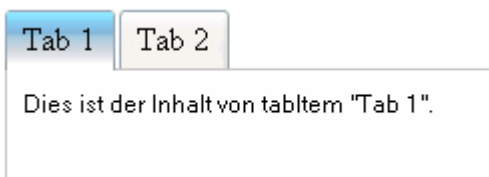


Abbildung 59: Das NetAdvantage for JSF-WebTab-Steuerelement

Die Registerkartenkomponente unterstützt eine Anpassung via CSS, Hovereffekte und deaktivierte Registerkarten.

5.12.3.2.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Zustand der Registerkarten-Komponente wird über Request-Grenzen hinweg persistiert. Bei deaktiviertem JavaScript ist jedoch ein Registerkartenwechsel nicht mehr möglich, weshalb von keiner Persistierungsunterstützung in diesem Fall mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Der Registerkarteninhalt lässt sich völlig frei durch die Platzierung und Konfiguration beliebiger JSF-Komponenten (innerhalb eines tabItem-Elements) anpassen. Die Registerkartenknöpfe können via CSS angepasst werden. Eine templatebasierte Anpassbarkeit besteht nicht; wird jedoch auch nicht vorausgesetzt. Neben der Anpassung der statischen Stile der Registerkartenkomponente kann auch die Darstellung des Selektionszustands und des Hoverzustands via CSS angepasst werden.

Bewertung: 1,0

Verhalten

Ein Klick auf einen Registerkartenknopf bewirkt die Anzeige des zugehörigen Registerkarteninhalts. Eine reine Bedienung mit der Tastatur ist über die Tab-Taste möglich. Accesskeys werden jedoch nicht unterstützt. Ein Anklicken einer Registerkarte oder eine Bedienung mit der Tastatur zeigt jedoch nur bei aktiviertem JavaScript einen Effekt, da ohne JavaScript die zugrundeliegenden Funktionen nicht ausgeführt werden.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung unter Eclipse 3.0 (mit WTP), IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt. Dabei können WebTab-Steuerelemente bzw. tabView- und tabItem-Komponenten via Drag & Drop positioniert und ihre Eigenschaften festgelegt werden. Untergeordnete Steuerelemente lassen sich ebenfalls über Drag & Drop auf den einzelnen Tabs positionieren, die in einer vereinfachten WYSIWYG-Ansicht des Designers dargestellt werden.

Bewertung: 1,0

5.12.3.2.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript funktioniert das WebTab-Steuerelement nicht mehr, da der Tabwechsel JavaScript erfordert.

Bewertung: 0,0

5.12.3.2.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, sind diese über die JSF-Bordmittel zu verwenden, sodass die Spezifikation alternativer Texte in der Verantwortung des jeweiligen Entwicklers liegt. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponente im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Eine Ausnahme bildet das Standard-Hintergrundmuster der Registerkartenknöpfe, welches aber flexibel angepasst werden kann. Die Darstellung ist grundsätzlich über CSS anpassbar.

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente in ihrer Gesamtheit sämtliche darstellungsrelevanten Eigenschaften über CSS angepasst werden können. Für die Tabinhalte zeichnet sich der jeweilige Entwickler verantwortlich, sodass auch hier prinzipiell keine Einschränkung besteht. Die Registerkartenknöpfe können flexibel angepasst werden, sodass auch hier die Möglichkeit besteht, via CSS Anpassungen an der Darstellung vorzunehmen.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung der Tab-Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von der Tab-Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Registerkarten basieren standardmäßig unter Anderem auf table-Elementen, was im Sinne der Barrierefreiheit als zweckentfremdend und folglich negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript führt dazu, dass kein Tabwechsel mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext der Registerkartensteuerelemente nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Die Registerkartenknöpfe können über die Tab-Taste fokussiert werden. Tastaturkürzel (Accesskeys) auf Tabknopfebene werden nicht unterstützt. Jedoch ist die bestehende Tastaturnavigierbarkeit nur verwendbar, wenn JavaScript aktiviert ist, da nur dann die zugeordneten Funktionen ausgeführt werden.

Bewertung: 0,5

Anforderung 10

Die Registerkartenkomponente setzt keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich der WebTab-Steuerelemente keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Registerkarten-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden jedoch nicht bereitgestellt. Zur verbesserten Orientierung und Navigation werden keine besonderen Strukturierungselemente verwendet, da stattdessen im Sinne der Barrierefreiheit negativ zu bewertende Tabellenlayouts eingesetzt werden.

Bewertung: 0,0

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.12.3.2.5 Browserinteroperabilität

Die Browserinteroperabilität stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.12.3.3 Menüs

5.12.3.3.1 Beschreibung

NetAdvantage for JSF ermöglicht die Realisierung von Menüs über die Komponente WebMenu (vgl. Infragistics, 2008b: 336-343), welches aus den Komponenten menu, menuItem,

menuItemCheckMark und menuItemSeparator besteht (vgl. ebd.: 336). Eine exemplarische Darstellung wird in Abbildung 60 aufgezeigt.

Es werden Separatoren, Radiobutton-Menüs, deaktivierte Menüs sowie diverse Möglichkeiten der Anpassung der Darstellung von Menüleiste und Menüelementen (auf und außerhalb der Menüleiste) unterstützt, wobei neben der Anpassbarkeit der regulären Darstellung auch Selektionszustände und Hoverzustände angepasst werden können. Menüs können neben einer horizontalen Darstellung auch vertikal angezeigt werden.

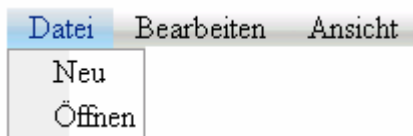


Abbildung 60: Die NetAdvantage for JSF-Menükomponenten

5.12.3.3.2 *Rich-Client-Funktionalität*

Anpassbarkeit

Durch Kombination von menu-, menuItem-, menuItemCheckMark- und menuItemSeparator-Komponenten können Menüs mit Text und Icons, Separatoren, Radiobutton-Menüs, deaktivierte Menüs sowie beliebig tiefe Menüstrukturen abgebildet werden, da menuItem-Komponenten verschachtelt werden können. Die Komponente menu fungiert dabei als Wurzelcontainer zur Aufnahme der Menüs der Menüleiste. Das Menü und dessen Bestandteile können völlig frei über CSS angepasst werden. Auch die weitergehende Stile wie Hovereffekte lassen sich damit anpassen. Die entsprechenden menüelementbezogenen Anpassungen lassen sich dabei auf globaler Ebene (menu-Steuerelement) oder auch auf Menüelementebene festlegen.

Bewertung: 1,0

Verhalten

Eine Bewegung des Mauszeigers auf einen Menüpunkt zeigt das entsprechende Untermenü an. Eine Bedienung mit der Tastatur ist über die Tab- und Cursorstasten möglich. Explizite Accesskeys-Unterstützung besteht jedoch nicht. Die Tastaturansteuerung funktioniert nur bei aktiviertem JavaScript.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung unter Eclipse 3.0 (mit WTP), IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt. Dabei können WebMenu-Steuerelemente bzw. die zugehörigen Komponenten via Drag & Drop positioniert und ihre Eigenschaften festgelegt werden.

Bewertung: 1,0

5.12.3.3.3 *Funktionsweise bei nicht-verfügbarem JavaScript*

Bei deaktiviertem JavaScript stellt das WebMenu-Control seine Funktionalität ein. Es werden keine Untermenüpunkte mehr dynamisch angezeigt, sodass auch die zugrundeliegenden Funktionen der entsprechenden Menüpunkte nicht mehr ausgeführt werden können.

Bewertung: 0,0

5.12.3.3.4 *Barrierefreiheit nach BITV*

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Falls welche eingesetzt werden, können Alternativtexte spezifiziert werden. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Durch CSS bestehen ferner umfassende Anpassungsmöglichkeiten.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche Möglichkeiten um globale (über die CSS-Eigenschaften des Steuerelements selbst) oder auch menüelementlokale (über die Eigenschaften der untergeordneten einzelnen Komponenten) CSS-Vorgaben anzuwenden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass auch für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

NetAdvantage-Menü-Komponenten basieren standardmäßig auf Html-Tabellenstrukturen, was im Sinne der Barrierefreiheit als negativ zu bewerten ist, da es eine semantische Zweckentfremdung darstellt.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass die Untermenüpunkte standardmäßig nicht mehr eingeblendet werden, wodurch eine Ausführung hinterlegter Funktionen auch nicht mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur (Tab- und Cursortasten) ist nur bei aktiviertem JavaScript auf sinnvolle Weise möglich.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt, da es sich standardmäßig um ein Tabellenlayout handelt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Als Elemente zur Strukturierung bzw. Verbesserung werden keine besonderen Strukturierungselemente eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.12.3.3.5 Browserinteroperabilität

Die Browserinteroperabilität stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.12.3.4 Symbolleisten

5.12.3.4.1 Beschreibung

Für Symbolleisten existiert keine dezidierte Komponente unter NetAdvantage for JSF. Es lässt sich jedoch ähnliche Funktionalität durch entsprechende Anpassung der bereits im Menüsteuerelementkontext untersuchten WebMenu-Komponente (vgl. Abbildung 61) erreichen. Damit lassen sich (bei Weglassen der Beschriftung) Symbolleistenbuttons sowie Buttons mit Kontextmenüs (durch untergeordnete Menüs) realisieren.



Abbildung 61: Das NetAdvantage for JSF-WebMenu-Steuerelement als Symbolleiste

5.12.3.4.2 Rich-Client-Funktionalität

Anpassbarkeit

Die unterstützten Buttonarten sind aufgrund der originär anderen Zielsetzung der Komponente beschränkt, da nur reguläre Buttons, Separatoren, Radiobuttons und Dropdowns unterstützt werden. Jedoch bestehen umfangreiche Visualisierungsanpassungen via CSS, was auf menu-globaler Ebene, oder aber auch auf menuItem-spezifischer Ebene erfolgen kann.

Bewertung: 1,0

Verhalten

Ein Klick auf ein Symbolleistenelement führt die hinterlegte Aktion aus bzw. erlaubt die Navigation zu einer hinterlegten Url. Eine Ansteuerung mit der Tastatur ist über die Tab-Taste, aber nicht über Accesskeys möglich. Die Komponente lässt sich jedoch nur bei aktiviertem JavaScript benutzen, da nur dann hinterlegte Aktionen ausgeführt werden.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung unter Eclipse 3.0 (mit WTP), IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt. Dabei können WebMenu-Steuerelemente bzw. die zugehörigen Komponenten via Drag & Drop positioniert und ihre Eigenschaften festgelegt werden.

Bewertung: 1,0

5.12.3.4.3 Funktionsweise bei nicht-verfügbarem JavaScript

Die Komponente funktioniert nur bei aktiviertem JavaScript.

Bewertung: 0,0

5.12.3.4.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Daneben verwendet keine Komponente standardmäßig Symbole. Zudem werden keine Image-Maps oder Multimedia-Präsentationen verwendet.

Bewertung: 1,0

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden.

Bewertung: 1,0

Anforderung 3

Es bestehen umfangreiche komponenten- bzw. elementeigenschaftenbasierende Möglichkeiten um globale oder auch symbolleistenelementlokale CSS-Vorgaben anzuwenden. Die globalen Vorgaben können direkt als Styleklassenangaben bei der menu-Komponente vorgegeben werden. Die lokalen Vorgaben sind über die Styleklassen der jeweiligen, dem Symbolleisten-Steuerelement untergeordneten, menuItem-Komponenten anpassbar.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von

Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Das Rendering erfolgt standardmäßig als Tabelle, was im Sinne der Barrierefreiheit als Zweckentfremdung und damit negativ zu bewerten ist.

Bewertung: 0,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript hat zur Folge, dass die den Symbolleistenschaltflächen zugrundeliegenden Aktionen nicht mehr ausgeführt werden.

Bewertung: 0,5

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine komfortable Ansteuerung (Selektion und Bestätigung) mit der Tastatur (Tab- und Cursortasten) ist nur bei aktiviertem JavaScript auf sinnvolle Weise möglich.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die Menü-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Es werden keine Elemente zur Verbesserung der Strukturierung und Navigation verwendet. Stattdessen wird auf ein Tabellenlayout gesetzt.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Zur Verbesserung der Strukturierung und Navigation werden Html-Tabellen statt Html-Listenelementen eingesetzt, was eine Zweckentfremdung darstellt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.12.3.4.5 Browserinteroperabilität

Die Browserinteroperabilität stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.12.3.5 Hierarchische Steuerelemente (Tree Views)

5.12.3.5.1 Beschreibung

Hierarchische Strukturen können unter NetAdvantage for JSF über die Komponente WebTree (vgl. Infragistics, 2008b: 351-353) abgebildet werden. Dabei besteht WebTree nach Infragistics (2008b: 351) aus den Komponenten treeView und treeNode, wodurch der Baum selbst und seine

Knoten modelliert werden können. Bemerkt werden soll, dass durch eine Verschachtelung von treeNode-Elementen prinzipiell beliebig tiefe hierarchische Strukturen aufgebaut werden können.

Die Komponente unterstützt AJAX-basiertes, automatisches Paging, wodurch unter Anderem auch automatisch Paging-Navigationsschaltflächen eingeblendet werden können. Die treeNode-Elemente erlauben die Spezifikation von Knotensymbolen und Knotenbeschriftungen. Für jeden Baumknoten lassen sich Icons für den kollabierten und den expandierten Zustand festlegen. Auch können Stile für den Hoverzustand spezifiziert werden. Neben der Ausführung von serverseitigem Code ist auch die Navigation zu einer anderen Webressource möglich.

Daneben existieren flexible Data Binding-Mechanismen. Unter Anderem werden auch rekursive Datenquellen unterstützt. Eine Darstellung der Komponente liegt in Abbildung 62 vor.



Abbildung 62: Die NetAdvantage for JSF-WebTree-Komponente

5.12.3.5.2 Rich-Client-Funktionalität

Request-Lebenszyklus-Persistenz

Der Expansionszustand des Steuerelements bleibt nach einem Submit erhalten, falls JavaScript aktiviert ist. Bei deaktiviertem bzw. nicht verfügbarem JavaScript können keine Baumknoten mehr expandiert oder kollabiert werden, sodass in diesem Fall auch von keiner echten Persistenz mehr gesprochen werden kann.

Bewertung: 0,5

Anpassbarkeit

Das Steuerelement ist flexibel anpassbar. Zum Einen kann der Komponente eine CSS-Klasse zugewiesen werden, über welche prinzipiell auch untergeordnete Elemente graphisch angepasst werden können. Zum Anderen kann auch jeweils auf Baumknotenebene das Erscheinungsbild des Baumknotens via CSS sowie der Spezifikation von anzuzeigendem Symbol und Text angepasst werden. Zudem wird die Spezifikation eines Hoverstils unterstützt.

Bewertung: 1,0

Verhalten

Unterelemente eines Baumknotens werden per Klick auf das Expansions- bzw. Kollabierungssymbol angezeigt bzw. verborgen, jedoch nur bei aktiviertem JavaScript. Ein Anklicken eines Baumknotens bewirkt in Abhängigkeit von den gesetzten Eigenschaften eine Navigation oder die Verarbeitung eines serverseitigen Ereignisses.

Bewertung: 0,5

Entwicklungsunterstützung

Es wird eine komfortable Entwicklungsunterstützung unter Eclipse 3.0 (mit WTP), IBM Rational Application Developer for WebSphere Software 7.0 und NetBeans 5.5 unterstützt. Dabei können

WebTree-Steuerelemente bzw. die zugehörigen Komponenten via Drag & Drop positioniert und ihre Eigenschaften festgelegt werden.

Bewertung: 1,0

5.12.3.5.3 Funktionsweise bei nicht-verfügbarem JavaScript

Bei deaktiviertem JavaScript stellt das TreeView-Steuerelement seine Funktionalität dahingehend ein, dass kein Expandieren bzw. Kollabieren von Baumknoten dann mehr möglich ist, wodurch auch kein Zugriff mehr auf untergeordnete Elemente möglich ist.

Bewertung: 0,0

5.12.3.5.4 Barrierefreiheit nach BITV

Anforderung 1

Keine der Komponenten gibt Audio- oder Videoinformationen aus. Bei Spezifikation von Baumknotensymbolen können jedoch keine spezifische Alternativtexte über die jeweils innerhalb eines Baumknoten platzierten Komponenten erhalten. Auch das Setzen eines Tooltips ermöglicht nicht die Umsetzung in einen alternativen Text.

Bewertung: 0,5

Anforderung 2

Da keine spezifische Darstellung zwingend vorgegeben wird, kann auch keinerlei negative Bewertung der Komponenten im Hinblick auf die Verständlichkeit bei Elimination des Aspekts Farbe vorgenommen werden. Die Wahl der Farben verbleibt somit in der Verantwortung des Entwicklers. Standardmäßig kann jedoch ein klassisches Windows Ordner-Symbol angezeigt werden, was jedoch problemlos ausgetauscht werden kann

Bewertung: 1,0

Anforderung 3

Zur Anforderung 3 ist zu bemerken, dass für die Komponente global graphische Einstellungen über die Komponenten-CSS-Klasse vorgenommen werden können. Auch auf Baumknotenebene können über die Baumknotensteuerelemente Anpassungen hinsichtlich der Visualisierung vorgenommen werden.

Bewertung: 1,0

Anforderung 4

Mehrsprachigkeit wird nicht explizit unterstützt, sodass für diesen Punkt kein Anwendungsbereich im Zusammenhang mit der Evaluierung dieser Komponente vorliegt. Die Festlegung von Akronymen und Abkürzungen liegt in der Zuständigkeit des Entwicklers und ist unabhängig von dieser Komponente zu betrachten.

Bewertung: 1,0

Anforderung 5

Es findet keine Zweckentfremdung des table-Elements bei dieser Komponente statt, da hierarchische Strukturen standardmäßig über span-Elemente aufgebaut werden.

Bewertung: 1,0

Anforderung 6

Bei Deaktivierung von CSS ist die Anwendung weiterhin benutzbar. Eine Deaktivierung von JavaScript bewirkt, dass die Komponente überhaupt nicht mehr verwendet werden kann, was bedeutet, dass keine Knotenkollabierung bzw. Knotenexpansion aufgrund fehlender Darstellung mehr möglich ist.

Bewertung: 0,0

Anforderung 7

Diese Anforderung ist im Kontext des evaluierten Steuerelements nicht anwendbar, da keine zeitgesteuerten Funktionalitäten vorliegen.

Bewertung: 1,0

Anforderung 8

Für Aspekt 8 gilt analog zu Aspekt 7, dass keine spezifische Anwendbarkeit vorliegt.

Bewertung: 1,0

Anforderung 9

Es werden keine gerätespezifischen Eventhandler oder anderweitigen Funktionen benutzt, die einen Einsatz des Frameworks auf ein bestimmtes Ein- oder Ausgabegerät beschränken. Eine Ansteuerung mit der Tastatur wird (von Accesskeys abgesehen) unterstützt. Bei deaktiviertem JavaScript ergibt dies jedoch keinen Sinn, da keine hinter den Knoten liegende Funktionalität auf unteren Ebenen mehr ausgeführt werden kann.

Bewertung: 0,5

Anforderung 10

Diese Komponenten setzen keine Popups ein.

Bewertung: 1,0

Anforderung 11

Für Anforderung 11 findet sich bezüglich des evaluierten Steuerelements keinerlei sinnvolle Anwendbarkeit.

Bewertung: 1,0

Anforderung 12

Die TreeView-Steuerelemente selbst machen von keinerlei Frames Gebrauch. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt, da standardmäßig auf Html-Span-Elemente gesetzt wird. Es gibt jedoch optional die Möglichkeit Ergebnisse eines Baumknotenklicks in einen anderen Frame zu leiten.

Bewertung: 0,5

Anforderung 13

Anforderung 13 ist im Kontext dieser Steuerelemente nicht relevant. Besondere Meta-Informationen werden nicht bereitgestellt. Bestimmte Elemente zur Strukturierung bzw. Verbesserung der Navigation werden nicht eingesetzt.

Bewertung: 0,5

Anforderung 14

Das Sprachniveau und die Darstellung der Texte liegen in der Hand des jeweiligen Entwicklers.

Bewertung: 1,0

5.12.3.5.5 Browserinteroperabilität

Die Browserinteroperabilität stellt sich wie folgt dar.

Internet Explorer 6.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Mozilla Firefox 2.0+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Opera 9+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

Safari 3.1+

Es liegen keine funktionalen Einschränkungen vor.

Bewertung: 1,0

5.12.3.6 Fortschrittsanzeige

Es existiert keine dezidierte Komponente im NetAdvantage for JSF-Portfolio, welche für den Anwendungsbereich der Statusindikation von laufenden Prozessen ausgelegt worden ist.

Bewertung: 0,0

5.12.3.7 Infotips

Es existiert keine dezidierte Komponente im NetAdvantage for JSF-Portfolio, welche für den Anwendungsbereich der Tooltip- bzw. Infotip-Unterstützung ausgelegt worden ist.

Bewertung: 0,0

5.12.4 Fazit

NetAdvantage for JSF stellt eine brauchbare Webapplikationsbibliothek-Lösung im Java EE-Bereich dar, wenngleich der Funktionsumfang noch hinter den untersuchten Konkurrenzprodukten, vor allem hinsichtlich der templatebasierten Anpassbarkeit, zurücksteht.

Wenngleich eine gute Anforderungserfüllung festzustellen ist, ist als absolutes Ausschlusskriterium im eGovernment-Kontext zu nennen, dass nahezu alle Komponenten nur bei verfügbarem und aktiviertem JavaScript auf Clientseite funktionieren.

6 Fazit

Im Rahmen dieser Untersuchung wurden die bedeutendsten und vielversprechendsten Frameworks im Java- und .NET-Webapplikationsumfeld im Hinblick auf die Aspekte Featurereichtum und Barrierefreiheit untersucht. Dabei lag der Fokus nicht auf der Menge an Rich Client-Funktionalität des jeweiligen Frameworks, sondern auf der Erfüllung der formulierten Anforderungen, die sich wiederum aus einer Voruntersuchung von Desktopentwicklungs-Guidelines ergeben haben.

Zentrale Rolle bei der Untersuchung spielte neben der Untersuchung der Barrierefreiheit im Allgemeinen der Aspekt JavaScript-Unabhängigkeit, da die meisten anderen Anforderungen aus dem Bereich Barrierefreiheit oftmals erfüllt sind, aber dieser Punkt meistens ignoriert oder ausgeblendet wird. Aus diesem Grund wurde der genannte Aspekt auch als notwendige Anforderung deklariert, was bedeutet, dass das entsprechende Framework diesen Aspekt erfüllen muss, um eine positive Gesamtbewertung zu erhalten.

Bei Betrachtung des Gesamtergebnis-Tabellenwerks (Anlage A) und der dazugehörigen Visualisierung (Anlage B) kann festgestellt werden, dass die durchschnittliche Anforderungserfüllung bei 47 % und die maximale Anforderungserfüllung bei 64 % (geleistet durch das Framework ASPxperience) liegt. Beide Werte liegen unter der geforderten Mindestanforderungserfüllung von 70 %.

Weitaus problematischer stellt sich das Ergebnis noch bei Betrachtung des Aspekts JavaScript-Unabhängigkeit dar. Nur drei von sieben Frameworks (43 %) berücksichtigen den Aspekt Barrierefreiheit und auch diese jeweils nur bei einer Komponente dahingehend, dass die jeweiligen Komponenten auch ohne JavaScript-Verfügbarkeit ihre Kernfunktionalität wahren. Dies ergibt bei Betrachtung aller Frameworks und deren Komponenten, wie in Anlage B dargestellt, einen Erfüllungsgrad dieser zentralen Anforderung von lediglich 4 %.

Zusammenfassend kann der Schluss gezogen werden, dass keines der untersuchten Frameworks gegenwärtig dazu im Stande ist, die besonderen Anforderungen, die sich im Bereich des eGovernment im Hinblick auf barrierefreie Rich Client-Applikationen ergeben, ausreichend zu befriedigen. Besonders im Hinblick auf den Aspekt der Unabhängigkeit von JavaScript liegen derzeit enorm große Defizite bei allen geprüften Lösungen vor. Diese ist jedoch nach Barrierefreiheitsrichtlinien wie der BITV von eminenter Bedeutung, sodass derzeit kein Webentwicklungsframework auf dem Markt erhältlich ist, welches den Richtlinien der BITV entspricht.

Auch die Kombination verschiedener Technologien, die in der Untersuchung bei den ersten evaluierten Frameworks bereits zum Vorteil der untersuchten Frameworks berücksichtigt worden ist, ist nicht immer problemlos möglich und ein Garant für die Erhöhung des Grads der Anforderungserfüllung.

Als Konsequenz wird vorgeschlagen, ein neues Framework von Grund auf zu konzeptionieren und umzusetzen, welches sich die Erfüllung der genannten Anforderungen im Bereich des eGovernment als oberstes Ziel setzt.

Literaturverzeichnis

[Adobe Systems Incorporated 2006] Adobe Systems Incorporated (Hrsg.): *Adobe Flex 2. Getting started with Flex 2*. San Jose, 2006. – URL http://download.macromedia.com/pub/documentation/en/flex/2/flex2_gettingstarted.pdf. – Letzter Zugriff am 17.06.2008

[Ananiev und Nourie 2006] Ananiev, Artem ; Nourie, Dana: The New Modality API in Java SE 6. 2006. – URL <http://java.sun.com/developer/technicalArticles/J2SE/Desktop/javase6/modality/>. – Letzter Zugriff am 17.06.2008

[Apple Inc. 2008] Apple Inc. (Hrsg.): *Apple Human Interface Guidelines*. 2008. – URL <http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/>. – Letzter Zugriff am 05.05.2008

[Bayerisches Landesamt für Finanzen 2008] Bayerisches Landesamt für Finanzen (Hrsg.): *IuK-Verfahren des LfF*. 2008. – URL Internes Dokument des Landesamts für Finanzen. – Letzter Zugriff am 02.07.2008

[Bayerisches Staatsministerium des Innern 2003] Bayerisches Staatsministerium des Innern (Hrsg.): *Bayerische Barrierefreie Informationstechnik-Verordnung (BayBITV)*. Juni 2003

[Benson et al. 2004] Benson, Calum ; Elman, Adam ; Nickell, Seth ; Robertson, Colin Z.: *GNOME Human Interface Guidelines 2.0*. 2004. – URL <http://developer.gnome.org/projects/gup/hig/2.0/hig-2.0.pdf>. – Letzter Zugriff am 18.02.2008

[Boehm 2006] Boehm, Barry: A view of 20th and 21st century software engineering. In: *ICSE '06: Proceedings of the 28th international conference on Software engineering*. New York, NY, USA: ACM, 2006, S. 12–29

[Brown 2000] Brown, Kevin: Keep stateless Web pages in mind when planning your Web site. 2000. – URL http://articles.techrepublic.com.com/5100-10878_11-5030231.html. – Letzter Zugriff am 09.05.2008

[Bundesministerium des Innern 2002] Bundesministerium des Innern (Hrsg.): *Barrierefreie Informationstechnik-Verordnung (BITV)*. Juli 2002. – URL <http://bundesrecht.juris.de/bundesrecht/bitv/gesamt.pdf>

[Debije-Meessen und Jansen 2006] Debije-Meessen, A. E. J. ; Jansen, J. A. H.: The balance between Aesthetics, Usability and Corporate Identity: Graphic User Interface design within a commercial company. In: *IV '06: Proceedings of the conference on Information Visualization*. Washington, DC, USA : IEEE Computer Society, 2006, S. 357–361

[Developer Express Inc. 2008a] Developer Express Inc. (Hrsg.): *ASPxperience Suite Overview*. 2008. – URL <http://www.devexpress.com/Products/NET/WebForms/ASPxperience/index.xml>. – Letzter Zugriff am 24.05.2008

[Developer Express Inc. 2008b] Developer Express Inc. (Hrsg.): *ASPxperience Suite - Specifications*. 2008. – URL <http://www.devexpress.com/Products/NET/WebForms/ASPxperience/info.xml>. – Letzter Zugriff am 24.05.2008

- [Dubinko 2003] Dubinko, Micah: *XForms Essentials*. Sebastopol, CA, USA : O'Reilly & Associates, Inc., 2003
- [Duffner 2003] Duffner, Robert: Portals unlock the knowledge that drives business value. 2003, S. 202–219
- [Esposito 2004] Esposito, Dino: *Introducing ASP.NET 2.0*. Redmond, WA, USA : Microsoft Press, 2004
- [Esposito 2005] Esposito, Dino: *Programming Microsoft® ASP.NET 2.0 Core Reference*. Microsoft Press Corp., 2005
- [Esposito 2007] Esposito, Dino: Brücken für Webservices: Externe Webservices in ASP.NET AJAX aufrufen. In: *dotnetpro* 6 (2007), S. 28–33
- [Esposito 2008] Esposito, Dino: Anpassen von Steuerelementen mit AJAX-Extendern. In: *MSDN Magazine* 1 (2008)
- [Gamma et al. 1994] Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, Massachusetts : Addison Wesley, 1994
- [Geary und Horstmann 2007] Geary, David M. ; Horstmann, Cay S.: *Core JavaServer Faces (Core)*. Prentice Hall International, 2007
- [Gisler et al. 2001] Gisler, Michael ; Günter, Matthias ; Spahni, Dieter: eGovernment. In: *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 5*. Washington, DC, USA : IEEE Computer Society, 2001, S. 5002
- [Infragistics 2008a] Infragistics (Hrsg.): *Infragistics Announces NetAdvantage for .NET 2008 Volume 1*. 2008. – URL <http://www.infragistics.com/Awesome.aspx?id=8583>. – Letzter Zugriff am 12.05.2008
- [Infragistics 2008b] Infragistics (Hrsg.): *NetAdvantage for JSF*. 2008. – URL http://dl-uklo.infragistics.com/products/NetAdvantage/JSF/2008.1/help/NetAdvantage_20081_JSF_Help_PDF.zip. – Letzter Zugriff am 05.06.2008
- [Jacobs und Kruse 2005] Jacobs, Dennis ; Kruse, Ulrike: Die Checkliste der BITV: Erläuterungen und Beispiele. 2005. – URL http://www.irs-basic.de/media/downloads/downloads_1/barrierefr_bitv-checkl-anm.pdf. – Letzter Zugriff am 09.05.2008
- [JBoss.org 2008a] JBoss.org (Hrsg.): *JBoss RichFaces*. 2008. – URL <http://www.jboss.org/jbossrichfaces/>. – Letzter Zugriff am 29.05.2008
- [JBoss.org 2008b] JBoss.org (Hrsg.): *RichFaces Developer Guide*. 2008. – URL <http://www.jboss.org/file-access/default/members/jbossrichfaces/freezone/docs/devguide/en/pdf/richfaces-usersguide.pdf>. – Letzter Zugriff am 30.05.2008
- [KBSt 2006a] KBSt (Hrsg.): *SAGA 3.0: Standards und Architekturen für E-Government-Anwendungen*. Oktober 2006. – URL http://www.kbst.bund.de/cln_012/nn_836802/SharedDocs/Anlagen-kbst/Saga/saga__3__0,templateId=raw,property=publicationFile.pdf/saga_3_0.pdf

- [KBSt 2006b] KBSt (Hrsg.): *V-Modell XT 1.2.1*. 2006. – URL <ftp://ftp.tu-clausthal.de/pub/institute/informatik/v-modell-xt/Releases/1.2.1/Dokumentation/V-Modell-XT-Gesamt.pdf>
- [KDE-Entwicklerteam o.J.] KDE-Entwicklerteam (Hrsg.): *KDE 3 Styleguide*. o.J.. – URL <http://developer.kde.org/documentation/standards/kde/style/styleguide.pdf>. – Letzter Zugriff am 18.02.2008
- [Linaje et al. 2007] Linaje, Marino ; Preciado, Juan C. ; Sánchez-Figueroa, Fernando: Engineering Rich Internet Application User Interfaces over Legacy Web Models. In: *IEEE Internet Computing* 11 (2007), Nr. 6, S. 53–59
- [Marinschek et al. 2006] Marinschek, Martin ; Müllan, Gerald ; Schnabl, Andrea: *JSF @ Work. JavaServer Faces und Apache MyFaces erfolgreich einsetzen*. Dpunkt Verlag, 2006
- [McClure et al. 2006] McClure, Wallace B. ; Cate, Scott ; Glavich, Paul ; Shoemaker, Craig: *Beginning Ajax with ASP.NET (Beginning)*. Birmingham, UK, UK : Wrox Press Ltd., 2006
- [Microsoft Corporation 2007a] Microsoft Corporation (Hrsg.): *Windows Vista User Experience Guidelines*. 2007. – URL <http://download.microsoft.com/download/e/1/9/e191fd8c-bce8-4dba-a9d5-2d4e3f3ec1d3/ux%20guide.pdf>. – Letzter Zugriff am 17.02.2008
- [Microsoft Corporation 2007b] Microsoft Corporation (Hrsg.): *Windows User Experience Guidelines. Windows User Experience Guidelines for Windows XP and Windows 2000*. 2007. – URL <http://www.ms2.cn/downloads/details.aspx?familyid=B996E1E7-A83A-4CAE-936B-2A9D94B11BC5&displaylang=en>. – Letzter Zugriff am 18.02.2008
- [Mitchell 2004] Mitchell, Scott: *Understanding ASP.NET View State*. 2004. – URL [http://msdn.microsoft.com/en-us/library/ms972976\(printer\).aspx](http://msdn.microsoft.com/en-us/library/ms972976(printer).aspx). – Letzter Zugriff am 08.05.2008
- [Moritz 2008] Moritz, Florian: *Rich Internet Applications (RIA): A Convergence of User Interface Paradigms of Web and Desktop Exemplified by JavaFX*, Fachhochschule Kaiserslautern, Diplomarbeit, Januar 2008. – URL <http://www.flomedia.de/diploma/documents/DiplomaThesisFlorianMoritz.pdf>. – Letzter Zugriff am 16.06.2008
- [Oracle 2008] Oracle (Hrsg.): *Web User Interface Developer's Guide for Oracle Application Development Framework, 11g Release 1: DRAFT 5/1/08*. 2008. – URL http://download.oracle.com/otn_hosted_doc/jdeveloper/11/doc/b31973.pdf. – Letzter Zugriff am 31.05.2008
- [Pemberton 2004] Pemberton, Steven (Hrsg.): *XHTML Frequently Answered XHTML Frequently Answered Questions*. 2004. – URL <http://www.w3.org/MarkUp/2004/xhtml-faq>. – Letzter Zugriff am 01.06.2008
- [Rader et al. 2007] Rader, Devin ; Beres, Jason ; Little, J. A. ; Hinkson, Grant: *Silverlight 1.0*. Birmingham, UK, UK : Wrox Press Ltd., 2007
- [SYS-CON Media Inc. 2008] SYS-CON Media Inc. (Hrsg.): *ComponentArt Releases Web.UI 2008.1: Featuring the New Hyper-Responsive AJAX UI*. 2008. – URL http://ajax.sys-con.com/read/529114_p.htm. – Letzter Zugriff am 11.05.2008

[Telerik 2008] Telerik (Hrsg.): *Why Choose RadControls for ASP.NET AJAX*. 2008. – URL <http://www.telerik.com/products/aspnet-ajax/why.aspx>. – Letzter Zugriff am 23.05.2008

[Vogel et al. 2005] Vogel, Oliver ; Arnold, Ingo ; Chughtai, Arif ; Völter, Markus: *Software-Architektur. Grundlagen - Konzepte - Praxis*. Spektrum Akademischer Verlag, 2005

[W3C 2008] W3C (Hrsg.): *Rich Web Clients Activity Statement*. 2008. – URL <http://www.w3.org/2006/rwc/Activity.html>. – Letzter Zugriff am 16.06.2008

[Wenz 2007] Wenz, Christian: *Programming ASP.NET AJAX*. 1. O'Reilly Media, 2007

Abbildungsverzeichnis

Abbildung 1: Definition des Grades der Gesamtanforderungserfüllung.....	15
Abbildung 2: Das ASP.NET-Menu-Steuerelement.....	18
Abbildung 3: Das ASP.NET-TreeView-Steuerelement.....	22
Abbildung 4: Das ASP.NET-UpdateProgress-Steuerelement.....	25
Abbildung 5: Das AJAX Control Toolkit-ConfirmButton-Control.....	30
Abbildung 6: Das AJAX Control Toolkit-ModalPopup-Control.....	30
Abbildung 7: Das AJAX Control Toolkit-PopupControl-Steuerelement	30
Abbildung 8: Das AJAX Control Toolkit-Tab-Control	35
Abbildung 9: Das AJAX Control Toolkit-HoverMenu-Control	38
Abbildung 10: Das AJAX Control Toolkit-DropDown-Control	38
Abbildung 11: Das ComponentArt Web.UI-Dialog-Steuerelement	46
Abbildung 12: Das ComponentArt Web.UI-TabStrip-Steuerelement	50
Abbildung 13: Das ComponentArt Web.UI-Menu-Control.....	54
Abbildung 14: Das ComponentArt Web.UI-ToolBar-Control.....	57
Abbildung 15: Das ComponentArt Web.UI-TreeView-Steuerelement	60
Abbildung 16: Das ComponentArt Web.UI-Dialog-Steuerelement als InfoTip.....	64
Abbildung 17: Das NetAdvantage-Dialog-Steuerelement.....	68
Abbildung 18: Das NetAdvantage-UltraWebTab-Steuerelement.....	73
Abbildung 19: Das NetAdvantage-UltraWebMenu-Control	76
Abbildung 20: Das NetAdvantage-UltraWebToolBar-Control	80
Abbildung 21: Das NetAdvantage-UltraWebTree-Steuerelement.....	83
Abbildung 22: Das RadWindow-Steuerelement	88
Abbildung 23: Das RadTabStrip-Steuerelement und das RadMultiPage-Steuerelement	92
Abbildung 24: Das RadMenu-Control	96
Abbildung 25: Das RadToolBar-Control	99
Abbildung 26: Das RadTreeView-Steuerelement.....	102
Abbildung 27: Die RadToolTip-Komponente	106
Abbildung 28: Das ASPxPopupControl-Steuerelement	110
Abbildung 29: Das ASPxTabControl-Steuerelement	114
Abbildung 30: Das ASPxMenu-Control	118
Abbildung 31: Das ASPxNavBar-Control	121
Abbildung 32: Die ASPxPopupControl-Komponente	125
Abbildung 33: Das CollapsiblePanel-Steuerelement als Dialog mit Link-Trigger.....	131
Abbildung 34: Das TabbedPane-Steuerelement.....	136
Abbildung 35: Das JSCookMenu-Control	139
Abbildung 36: Das PanelNavigation2-Steuerelement als Menü.....	140
Abbildung 37: Das PanelNavigation2-Steuerelement als Symbolleiste	143
Abbildung 38: Die Tree2-Komponente.....	147
Abbildung 39: Die Tomahawk-Popup-Komponente	151
Abbildung 40: Das Rich Faces-ModalPanel-Steuerelement.....	155
Abbildung 41: Das RichFaces-TabPanel-Steuerelement	160
Abbildung 42: Das RichFaces-DropDownMenu-Control.....	164
Abbildung 43: Das RichFaces-ContextMenu-Control.....	164
Abbildung 44: Das RichFaces-ToolBar-Steuerelement.....	167
Abbildung 45: Die RichFaces-Tree-Komponente.....	171
Abbildung 46: Das RichFaces-ProgressBar-Steuerelement.....	175
Abbildung 47: Exemplarischer Einsatz des RichFaces-Status-Steuerelements	175
Abbildung 48: Die RichFaces-ToolTip-Komponente.....	179
Abbildung 49: Das ADF Faces-Dialog-Steuerelement.....	184

Abbildung 50: Das ADF Faces-PanelWindow-Steuerelement	184
Abbildung 51: Das ADF Faces-PanelTabbed-Steuerelement	189
Abbildung 52: Das ADF Faces-PanelAccordion-Steuerelement	189
Abbildung 53: Die ADF Faces-Menükomponenten	193
Abbildung 54: Das ADF Faces-Toolbar-Steuerelement in einer Toolbox-Komponente.....	196
Abbildung 55: Die ADF Faces-Tree-Komponente	200
Abbildung 56: Das ADF Faces-ProgressIndicator-Steuerelement.....	204
Abbildung 57: Die ADF Faces-Popup-Komponente als Tooltip	207
Abbildung 58: Das NetAdvantage for JSF-WebDialogWindow-Steuerelement.....	212
Abbildung 59: Das NetAdvantage for JSF-WebTab-Steuerelement	216
Abbildung 60: Die NetAdvantage for JSF-Menükomponenten.....	220
Abbildung 61: Das NetAdvantage for JSF-WebMenu-Steuerelement als Symbolleiste.....	223
Abbildung 62: Die NetAdvantage for JSF-WebTree-Komponente	227

Die Abbildungen resultieren größtenteils auf selbst erstellten Screenshots eigener Testanwendungen zu den jeweiligen Frameworks bzw. auf ausschnittweisen Screenshots zu vom Hersteller bereitgestellten Testanwendungen. Sämtliche Rechte an den in den Abbildungen dargestellten Komponenten, sowie alle Rechte an Warenzeichen und Produktnamen liegen beim jeweiligen Eigentümer.

Anlagen

A. Gesamtergebnisse (Tabellenwerk)

Nr.	Framework	Komponenten (Anforderungserfüllung in %)							Durchschnittl. Anford.erf.	Notw. Vorauss. (% Grad)	
		Dial- oge	Register- karten	Menüs	Symbol- leisten	Hierarch. Komp.	Fortschritt s- anzei- ge	Infotips			
1	ASP.NET 3.5 (Standard)	0,00	0,00	0,62	0,00	0,60	0,72	0,00	0,28	0,00	0,43
2	AJAX Control Toolkit	0,49	0,61	0,58	0,00	0,57	0,72	0,62	0,51	0,00	0,86
3	ComponentArt Web.UI	0,49	0,66	0,65	0,65	0,67	0,72	0,58	0,63	0,00	1,00
4	NetAdvantage for ASP.NET	0,54	0,62	0,52	0,65	0,67	0,72	0,00	0,53	0,00	0,86
5	Telerik Controls f. ASP.NET AJAX	0,57	0,66	0,68	0,69	0,66	0,72	0,00	0,57	0,00	0,86
6	DevExpress ASPxperience	0,56	0,63	0,65	0,60	0,60	0,72	0,69	0,64	0,00	1,00
7	JSF (Standard)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
8	MyFaces Tomahawk	0,54	0,89	0,59	0,65	0,63	0,00	0,62	0,56	0,14	0,86
9	JBoss RichFaces	0,54	0,63	0,62	0,92	0,60	0,00	0,62	0,56	0,14	0,86
10	Oracle ADF Faces Rich Client	0,40	0,49	0,48	0,65	0,50	0,00	0,51	0,43	0,14	0,86
11	NetAdvantage for JSF	0,51	0,62	0,65	0,65	0,63	0,00	0,00	0,44	0,00	0,71

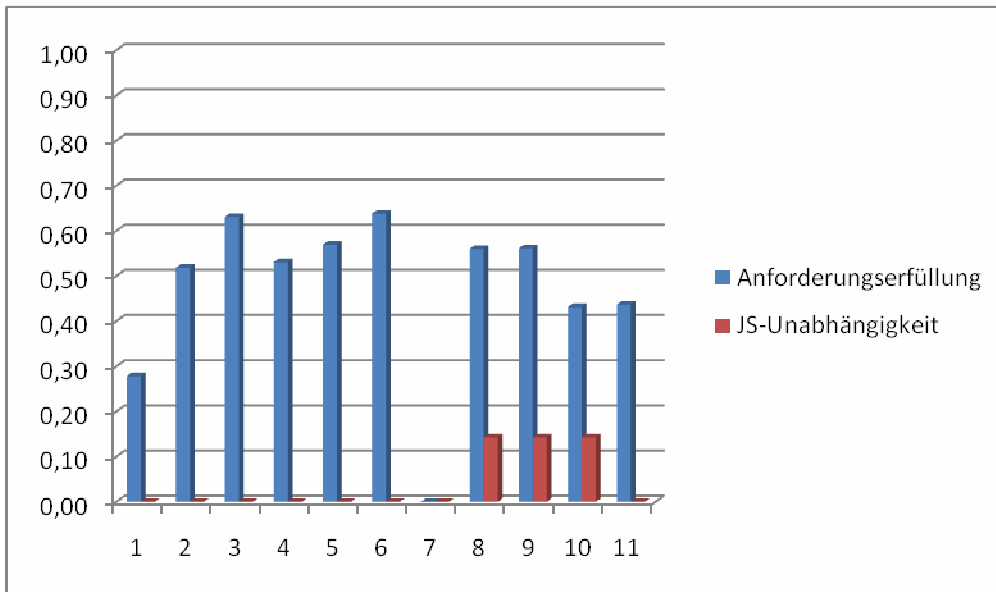
Kennzahlen

JS-Unabhängigkeit Proz. Anteil der Frameworkkomponenten, die auch ohne JavaScript zumindest ihre Kernfunktionalität wahren
Komponentenabdeckung Proz. Anteil d. im Rahmen d. Untersuchung geforderten Komponenten, die Bestandteil des Frameworks sind
Durchschnitt. Anforderungserfüllung Durchschnittliche Anforderungserfüllung bei gleichgewichteten Anforderungen

Notwendige Anforderungen

Durchschnittliche Anforderungserfüllung > 70%
JS-Unabhängigkeit > 90%
Komponentenabdeckung > 90%

B. Gesamtergebnisse (Visualisierung zum Tabellenwerk)



Horizontale Achse: Durchnummerierte Liste aller Frameworks

Vertikale Achse: Grad der Anforderungserfüllung (1,00 entspricht 100%)

Ergebnis:

- Mittlere Anforderungserfüllung: **47 %**
- Maximale Anforderungserfüllung: **64 % (DevExpress ASPxperience)**
- JavaScript-Unabhängigkeit
 - **Nur 3 der 11 getesteten Frameworks besitzen Komponenten, welche ohne JS-Verfügbarkeit ihre Kernfunktion wahren** – der Rest ist ohne JS nicht sinnvoll verwendbar
 - **Die 3 besagten Frameworks beinhalten jeweils nur 1 von 7 geprüften Komponenten mit JavaScript-Unabhängigkeit** -> 14% der Komponenten dieser Frameworks sind ohne JavaScript einsetzbar
 - **Insgesamt: (14% von 3/11 Frameworks) Komponenten, damit nur 4%, aller geprüften Komponenten, sind im Sinne der Barrierefreiheit einsetzbar (wenn man die restlichen verletzten Barrierefreiheitsanforderungen außer Acht lässt)**

C. [ASP.NET 3.5]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz					0,50		
	Anpassbarkeit			1,00		0,50	1,00	
	Verhalten			0,50		0,50	1,00	

	Entwicklungs- unterstützung			1,00		1,00	1,00	
	Wiederverwend- barkeit (ext. Res.)							
	Definierter Daten- austauschprozess					-	-	
	Standard-Dialoge							
	Durchschnitt			0,83		0,63	1,00	
Funktionsweise bei nicht ver- fügbarem JS	Funktionsweise bei nicht verfügba-rem JavaScript			0,00		0,00	0,00	
	Durchschnitt			0,00		0,00	0,00	
BITV-Anforde- rungserfüllung	Anforderung 1			1,00		1,00	1,00	
	Anforderung 2			1,00		1,00	1,00	
	Anforderung 3			0,50		0,50	1,00	
	Anforderung 4			1,00		1,00	1,00	
	Anforderung 5			0,50		0,50	1,00	
	Anforderung 6			0,00		0,00	0,00	
	Anforderung 7			1,00		1,00	1,00	
	Anforderung 8			1,00		1,00	1,00	
	Anforderung 9			1,00		1,00	1,00	
	Anforderung 10			1,00		1,00	1,00	
	Anforderung 11			1,00		1,00	1,00	
	Anforderung 12			0,50		0,50	1,00	
	Anforderung 13			0,50		0,50	0,50	
	Anforderung 14			1,00		1,00	1,00	
	Durchschnitt			0,79		0,79	0,89	
Browserinter- operabilität	Internet Explorer 6.0+			1,00		1,00	1,00	
	Firefox 2.0+			1,00		1,00	1,00	
	Opera 9			1,00		1,00	1,00	
	Safari 3.1			0,50		1,00	1,00	
	Durchschnitt			0,88		1,00	1,00	
Gesamt	Durchschnitt	0,00	0,00	0,62	0,00	0,60	0,72	0,00

D. [ASP.NET 3.5+AJAX-Control-Toolkit]-Rich Client- Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Register- karten	Menüs	Symbol- leisten	Hierarch. Steuerele- mente	Fortschritts- anzeige	Infotips
Rich Client- Funktionalität	Request-Lebens- zyklus-Persistenz	0,00	0,50			0,50	-	
	Anpassbarkeit	0,50	0,50	0,50		0,50	1,00	0,50
	Verhalten	0,00	0,50	0,50		0,50	1,00	0,50
	Entwicklungs- unterstützung	0,50	1,00	0,50		1,00	1,00	1,00
	Wiederverwend- barkeit (ext. Res.)	0,00				0,00	-	
	Definierter Daten- austauschprozess	0,00				-	-	
	Standard-Dialoge	0,00				-	-	
	Durchschnitt	0,14	0,63	0,50		0,50	1,00	0,67
Funktionsweise bei nicht verfüg-barem JavaScript	Funktionsweise bei nicht verfügba-rem JavaScript	0,00	0,00	0,00		0,00	0,00	0,00
	Durchschnitt	0,00	0,00	0,00		0,00	0,00	0,00
BITV-Anforde- rungserfüllung	Anforderung 1	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 2	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 3	1,00	1,00	1,00		0,50	1,00	1,00
	Anforderung 4	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 5	1,00	1,00	1,00		0,50	1,00	1,00
	Anforderung 6	0,00	0,00	0,00		0,00	0,00	0,00
	Anforderung 7	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 8	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 9	0,50	0,50	0,50		1,00	1,00	0,50
	Anforderung 10	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 11	1,00	1,00	1,00		1,00	1,00	1,00
	Anforderung 12	0,50	0,50	0,50		0,50	1,00	0,50
	Anforderung 13	0,50	0,50	0,50		0,50	0,50	0,50

	Anforderung 14	1,00	1,00	1,00		<u>1,00</u>	<u>1,00</u>	1,00
	Durchschnitt	0,82	0,82	0,82		0,79	0,89	0,82
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00		<u>1,00</u>	<u>1,00</u>	1,00
	Firefox 2.0+	1,00	1,00	1,00		<u>1,00</u>	<u>1,00</u>	1,00
	Opera 9	1,00	1,00	1,00		<u>1,00</u>	<u>1,00</u>	1,00
	Safari 3.1	1,00	1,00	1,00		<u>1,00</u>	<u>1,00</u>	1,00
	Durchschnitt	1,00	1,00	1,00		1,00	1,00	1,00
Gesamt	Durchschnitt	0,49	0,61	0,58	0,00	0,57	0,72	0,62
								0,51

E. [ASP.NET 3.5+ComponentArt Web.UI]-Rich Client- Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,00	1,00			0,50	-	
	Anpassbarkeit	0,50	1,00	1,00	1,00	1,00	<u>1,00</u>	0,50
	Verhalten	0,00	0,50	0,50	0,50	1,00	<u>1,00</u>	0,50
	Entwicklungsunterstützung	0,00	1,00	1,00	1,00	1,00	<u>1,00</u>	0,50
	Wiederverwendbarkeit (ext. Res.)	0,50					-	
	Definierter Datenaustauschprozess	0,00					-	
	Standard-Dialoge	0,00					-	
	Durchschnitt	0,14	0,88	0,83	0,83	0,88	1,00	0,50
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Durchschnitt	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 5	1,00	0,00	0,00	0,00	0,00	<u>1,00</u>	1,00

	Anforderung 6	0,00	0,00	0,00	0,00	0,00	<u>0,00</u>	0,00
	Anforderung 7	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 9	0,50	0,50	0,50	1,00	1,00	<u>1,00</u>	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 12	0,50	0,50	0,50	0,50	0,50	<u>1,00</u>	0,50
	Anforderung 13	0,50	0,50	0,50	0,50	0,50	<u>0,50</u>	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Durchschnitt	0,82	0,75	0,75	0,79	0,79	0,89	0,82
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Opera 9	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Safari 3.1	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Durchschnitt	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Gesamt	Durchschnitt	0,49	0,66	0,65	0,65	0,67	0,72	0,58
								<u>0,63</u>

F. [ASP.NET 3.5+NetAdvantage for ASP.NET]-Rich Client- Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,50	0,50			0,50	-	
	Anpassbarkeit	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Verhalten	0,00	0,50	0,50	0,50	1,00	<u>1,00</u>	
	Entwicklungsunterstützung	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Wiederverwendbarkeit (ext. Res.)	0,00					-	
	Definierter Datenaustauschprozess	0,00					-	
	Standard-Dialoge	0,00					-	

	Durchschnitt	0,36	0,75	0,83	0,83	0,88	1,00	
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,00	0,00	<u>0,00</u>	
	Durchschnitt	0,00	0,00	0,00	0,00	0,00	0,00	
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 5	0,50	0,00	0,00	0,00	1,00	<u>1,00</u>	
	Anforderung 6	0,00	0,00	0,00	0,00	0,00	<u>0,00</u>	
	Anforderung 7	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 9	0,50	0,50	0,50	0,50	0,50	<u>1,00</u>	
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Anforderung 12	0,50	0,00	0,50	0,50	0,50	<u>1,00</u>	
	Anforderung 13	0,50	0,50	0,50	0,50	0,50	<u>0,50</u>	
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Durchschnitt	0,79	0,71	0,75	0,75	0,82	0,89	
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	
	Opera 9	1,00	1,00	0,00	1,00	1,00	<u>1,00</u>	
	Safari 3.1	1,00	1,00	0,00	1,00	1,00	<u>1,00</u>	
	Durchschnitt	1,00	1,00	0,50	1,00	1,00	1,00	
Gesamt	Durchschnitt	0,54	0,62	0,52	0,65	0,67	0,72	0,00
								<u>0,53</u>

G. [ASP.NET 3.5+Telerik Controls for ASP.NET AJAX]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,00	0,50			0,50	-	
	Anpassbarkeit	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	0,50
	Verhalten	0,50	0,50	0,50	0,50	0,50	<u>1,00</u>	1,00
	Entwicklungsunterstützung	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Wiederverwendbarkeit (ext. Res.)	0,00					-	
	Definierter Datenaustauschprozess	0,50					-	
	Standard-Dialoge	0,50					-	
	Durchschnitt	0,50	0,75	0,83	0,83	0,75	1,00	0,83
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,00	0,00	<u>0,00</u>	0,00
	Durchschnitt	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	0,50	<u>1,00</u>	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 5	0,50	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 6	0,00	0,00	0,00	0,00	0,00	<u>0,00</u>	0,00
	Anforderung 7	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 9	0,50	0,50	0,50	1,00	1,00	<u>1,00</u>	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Anforderung 12	0,50	1,00	1,00	1,00	1,00	<u>1,00</u>	0,50
	Anforderung 13	0,50	1,00	1,00	1,00	1,00	<u>0,50</u>	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00

	Durchschnitt	0,79	0,89	0,89	0,93	0,89	0,89	0,82
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Opera 9	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Safari 3.1	1,00	1,00	1,00	1,00	1,00	<u>1,00</u>	1,00
	Durchschnitt	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Gesamt	Durchschnitt	0,57	0,66	0,68	0,69	0,66	0,72	0,00
								<u>0,57</u>

H. [ASP.NET 3.5+ASPxperience]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,50	0,50			0,50		
	Anpassbarkeit	1,00	1,00	1,00	0,50	<u>0,50</u>	<u>1,00</u>	1,00
	Verhalten	0,50	0,50	0,50	0,50	<u>0,50</u>	<u>1,00</u>	1,00
	Entwicklungsunterstützung	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Wiederverwendbarkeit (ext. Res.)	0,50				-	-	
	Definierter Datenaustauschprozess	0,00				-	-	
	Standard-Dialoge	0,00				-	-	
	Durchschnitt	0,50	0,75	0,83	0,67	0,63	1,00	1,00
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,00	<u>0,00</u>	<u>0,00</u>	0,00
	Durchschnitt	0,00	0,00	0,00	0,00	0,00	0,00	0,00
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	<u>0,50</u>	<u>1,00</u>	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 5	0,00	0,00	0,00	0,00	<u>0,50</u>	<u>1,00</u>	0,00
	Anforderung 6	0,00	0,00	0,00	0,00	<u>0,00</u>	<u>0,00</u>	0,00

	Anforderung 7	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 9	0,50	0,50	0,50	0,50	<u>1,00</u>	<u>1,00</u>	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Anforderung 12	0,50	0,50	0,50	0,50	<u>0,50</u>	<u>1,00</u>	0,50
	Anforderung 13	0,50	0,50	0,50	0,50	<u>0,50</u>	<u>0,50</u>	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Durchschnitt	0,75	0,75	0,75	0,75	0,79	0,89	0,75
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Firefox 2.0+	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Opera 9	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Safari 3.1	1,00	1,00	1,00	1,00	<u>1,00</u>	<u>1,00</u>	1,00
	Durchschnitt	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Gesamt	Durchschnitt	0,56	0,63	0,65	0,60	0,60	0,72	0,69
								<u>0,64</u>

I. [JSF (Standard)]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz					-	-	
	Anpassbarkeit					-	-	
	Verhalten					-	-	
	Entwicklungsunterstützung					-	-	
	Wiederverwendbarkeit (ext. Res.)						-	
	Definierter Datenaustauschprozess						-	
	Standard-Dialoge						-	
	Durchschnitt							

Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript							
	Durchschnitt					-	-	
BITV-Anforderungserfüllung	Anforderung 1					-	-	
	Anforderung 2					-	-	
	Anforderung 3					-	-	
	Anforderung 4					-	-	
	Anforderung 5					-	-	
	Anforderung 6					-	-	
	Anforderung 7					-	-	
	Anforderung 8					-	-	
	Anforderung 9					-	-	
	Anforderung 10					-	-	
	Anforderung 11					-	-	
	Anforderung 12					-	-	
	Anforderung 13					-	-	
	Anforderung 14					-	-	
	Durchschnitt							
Browserinteroperabilität	Internet Explorer 6.0+					-	-	
	Firefox 2.0+					-	-	
	Opera 9					-	-	
	Safari 3.1					-	-	
	Durchschnitt					-	-	
Gesamt	Durchschnitt	0,00	0,00	0,00	0,00	0,00	0,00	0,00
								0,00

J. [MyFaces Tomahawk]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Register- karten	Menüs	Symbol- leisten	Hierarch. Steuerele- mente	Fortschritts- anzeige	Infotips
Rich Client-Funktionalität	Request-Lebens- zyklus-Persistenz	0,50	1,00			0,50	-	
	Anpassbarkeit	0,50	0,50	0,50	1,00	1,00	-	1,00
	Verhalten	0,00	0,50	0,50	0,50	1,00	-	0,50
	Entwicklungs- unterstützung	1,00	1,00	0,50	0,50	0,50	-	0,50
	Wiederverwend- barkeit (ext. Res.)	0,00					-	
	Definierter Daten- austauschprozess	0,00					-	
	Standard-Dialoge	0,00					-	
	Durchschnitt	0,29	0,75	0,50	0,67	0,75	-	0,67
Funktionsweise bei nicht verfüg-barem JavaScript	Funktionsweise bei nicht verfügba-rem JavaScript	0,00	1,00	0,00	0,00	0,00	-	0,00
	Durchschnitt	0,00	1,00	0,00	0,00	0,00	-	0,00
BITV-Anforde- rungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 5	1,00	0,00	1,00	1,00	0,00	-	1,00
	Anforderung 6	0,00	1,00	0,00	0,00	0,00	-	0,00
	Anforderung 7	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 9	0,50	0,50	0,50	1,00	1,00	-	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	-	1,00
	Anforderung 12	1,00	0,50	1,00	1,00	0,50	-	0,50
	Anforderung 13	0,50	0,50	0,50	1,00	0,50	-	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	-	1,00

	Durchschnitt	0,86	0,82	0,86	0,93	0,79		0,82
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00		1,00
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00		1,00
	Opera 9	1,00	1,00	1,00	1,00	1,00		1,00
	Safari 3.1	1,00	1,00	1,00	1,00	1,00		1,00
	Durchschnitt	1,00	1,00	1,00	1,00	1,00		1,00
Gesamt	Durchschnitt	0,54	0,89	0,59	0,65	0,63	0,00	0,62
								0,56

K. [JBoss RichFaces]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,00	0,50			0,50		
	Anpassbarkeit	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Verhalten	0,00	0,50	0,50	0,50	0,00	1,00	0,50
	Entwicklungsunterstützung	1,00	1,00	0,50	1,00	1,00	1,00	0,50
	Wiederverwendbarkeit (ext. Res.)	0,00						
	Definierter Datenaustauschprozess	0,00						
	Standard-Dialoge	0,00						
	Durchschnitt	0,29	0,75	0,67	0,83	0,63	1,00	0,67
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	1,00	0,00	0,00	0,00
	Durchschnitt	0,00	0,00	0,00	1,00	0,00	0,00	0,00
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 5	0,50	0,00	1,00	0,00	0,00	1,00	1,00
	Anforderung 6	0,00	0,00	0,00	1,00	0,00	0,00	0,00

	Anforderung 7	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 9	1,00	0,50	0,50	1,00	1,00	1,00	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 12	1,00	0,50	0,50	0,50	0,50	1,00	0,50
	Anforderung 13	0,50	0,50	0,50	0,50	0,50	0,50	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Durchschnitt	0,86	0,75	0,82	0,86	0,79	0,89	0,82
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Opera 9	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Safari 3.1	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Durchschnitt	1,00	1,00	1,00	1,00	1,00	1,00	1,00
Gesamt	Durchschnitt	0,54	0,63	0,62	0,92	0,60	0,00	0,62
								0,56

L. [Oracle ADF Faces]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,00	0,50			0,50		
	Anpassbarkeit	1,00	1,00	1,00	1,00	1,00	0,50	1,00
	Verhalten	0,00	0,50	0,50	0,50	0,50	1,00	0,50
	Entwicklungsunterstützung	1,00	1,00	0,50	1,00	1,00	1,00	1,00
	Wiederverwendbarkeit (ext. Res.)	0,00						
	Definierter Datenaustauschprozess	0,00						
	Standard-Dialoge	0,50						
	Durchschnitt	0,36	0,75	0,67	0,83	0,75	0,83	0,83

Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,50	0,00	0,00	0,00
	Durchschnitt	0,00	0,00	0,00	0,50	0,00	0,00	0,00
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 2	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 3	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 4	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 5	0,50	1,00	0,00	0,00	1,00	1,00	1,00
	Anforderung 6	0,00	0,00	0,00	0,50	0,00	0,00	0,00
	Anforderung 7	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 8	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 9	1,00	0,50	0,50	0,50	1,00	1,00	0,50
	Anforderung 10	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 11	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Anforderung 12	1,00	0,50	0,50	0,50	0,50	1,00	0,50
	Anforderung 13	0,50	0,50	0,50	0,50	0,50	0,50	0,50
	Anforderung 14	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Durchschnitt	0,86	0,82	0,75	0,79	0,86	0,89	0,82
Browserinteroperabilität	Internet Explorer 6.0+	0,50	0,50	0,50	0,50	0,50	0,50	0,50
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Opera 9	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Safari 3.1	0,00	0,00	0,50	0,50	0,00	0,00	0,00
	Durchschnitt	0,38	0,38	0,50	0,50	0,38	0,38	0,38
Gesamt	Durchschnitt	0,40	0,49	0,48	0,65	0,50	0,00	0,51
								0,43

M. [NetAdvantage for JSF]-Rich Client-Barrierefreiheit-Anforderungsrealisierung

Prüfaspekt	Prüfsubaspekt	Dialoge	Registerkarten	Menüs	Symbolleisten	Hierarch. Steuerelemente	Fortschrittsanzeige	Infotips
Rich Client-Funktionalität	Request-Lebenszyklus-Persistenz	0,00	0,50			0,50		
	Anpassbarkeit	1,00	1,00	1,00	1,00	1,00		
	Verhalten	0,00	0,50	0,50	0,50	0,50		
	Entwicklungsunterstützung	0,50	1,00	1,00	1,00	1,00		
	Wiederverwendbarkeit (ext. Res.)	0,00						
	Definierter Datenaustauschprozess	0,00						
	Standard-Dialoge	0,00						
	Durchschnitt	0,21	0,75	0,83	0,83	0,75		
Funktionsweise bei nicht verfügbarem JavaScript	Funktionsweise bei nicht verfügbarem JavaScript	0,00	0,00	0,00	0,00	0,00		
	Durchschnitt	0,00	0,00	0,00	0,00	0,00		
BITV-Anforderungserfüllung	Anforderung 1	1,00	1,00	1,00	1,00	0,50		
	Anforderung 2	1,00	1,00	1,00	1,00	1,00		
	Anforderung 3	1,00	1,00	1,00	1,00	1,00		
	Anforderung 4	1,00	1,00	1,00	1,00	1,00		
	Anforderung 5	0,00	0,00	0,00	0,00	1,00		
	Anforderung 6	0,00	0,00	0,00	0,50	0,00		
	Anforderung 7	1,00	1,00	1,00	1,00	1,00		
	Anforderung 8	1,00	1,00	1,00	1,00	1,00		
	Anforderung 9	1,00	0,50	0,50	0,50	0,50		
	Anforderung 10	1,00	1,00	1,00	1,00	1,00		
	Anforderung 11	1,00	1,00	1,00	1,00	1,00		
	Anforderung 12	1,00	0,50	0,50	0,50	0,50		
	Anforderung 13	0,50	0,00	0,50	0,50	0,50		
	Anforderung 14	1,00	1,00	1,00	1,00	1,00		

	Durchschnitt	0,82	0,71	0,75	0,79	0,79		
Browserinteroperabilität	Internet Explorer 6.0+	1,00	1,00	1,00	1,00	1,00		
	Firefox 2.0+	1,00	1,00	1,00	1,00	1,00		
	Opera 9	1,00	1,00	1,00	1,00	1,00		
	Safari 3.1	1,00	1,00	1,00	1,00	1,00		
	Durchschnitt	1,00	1,00	1,00	1,00	1,00		
Gesamt	Durchschnitt	0,51	0,62	0,65	0,65	0,63	0,00	0,00
								0,44